Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Комсомольский-на-Амуре государственный технический университет»

М. Г. Некрасова

ДИСКРЕТНАЯ МАТЕМАТИКА

Часть 2

ТЕОРИЯ ГРАФОВ, ЭЛЕМЕНТЫ ТЕОРИИ КОНЕЧНЫХ АВТОМАТОВ

Утверждено в качестве учебного пособия
Ученым советом Федерального государственного бюджетного
образовательного учреждения высшего профессионального образования
«Комсомольский-на-Амуре государственный технический университет»

Комсомольск-на-Амуре
2013

УДК 519.17(07)
ББК 22.174я7
Н48

Рецензенты:

С. Б. Вениг, д-р физ.-мат. наук, профессор, декан факультета нано- и биомедицинских технологий ФГБОУ ВПО НИУ «Саратовский государственный университет им. Н.Г. Чернышевского»;
кафедра информационных систем, компьютерных технологий и физики ФГБОУ ВПО «Амурский гуманитарно-педагогический государственный университет», зав. кафедрой канд. физ.-мат. наук, доцент М. А. Савунов

Некрасова, М. Г.

Н48 Дискретная математика : учеб. пособие : в 2 ч. / М. Г. Некрасова. – Комсомольск-на-Амуре : ФГБОУ ВПО «КнАГТУ», 2013.

ISBN 978-5-7765-1064-9

Ч. 2. Теория графов, элементы теории конечных автоматов / М. Г. Некрасова.– 108 с.

ISBN 978-5-7765-1049-6

В учебном пособии изложены основные положения, типы графов, способы их задания, алгоритмы над графами, которые имеют практическое приложение для решения экономических задач. Рассмотрено сетевое планирование: понятие сетевого графика, критического пути и способы его нахождения. Даются определение конечных автоматов, способы их описания и анализа.

Приведены задания к РГЗ (10 вариантов) по всему курсу «Дискретная математика» и пример его выполнения.

Учебное пособие предназначено для студентов направлений 080500.62 – «Бизнес-информатика», 230700.62 – «Прикладная информатика» и специальности 080101.65 – «Экономическая безопасность».

УДК 519.17(07)
ББК 22.174я7

ISBN 978-5-7765-1049-6 (ч. 2)
ISBN 978-5-7765-1064-9

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Пособие по курсу «Дискретная математика» состоит из двух частей. В первой части рассмотрены элементы теории множеств и отношений, математическая логика. Во второй части рассматриваются теория графов, элементы теории конечных автоматов.

В рамках курса проходит знакомство с новыми средствами конструктивного анализа и моделирования в управлении – методами формального представления реальных управленческих ситуаций, процессов, систем. Поскольку пособие рассчитано на студентов, обучающихся по направлениям «Бизнес-информатика», «Прикладная информатика (в экономике)», «Экономическая безопасность», то упор сделан на постановку и решение прикладных задач. Для обеспечения эффективной самостоятельной работы студентов пособие дополнено практическими задачами и примерами, для каждого из изучаемых разделов автором разработаны тестовые материалы, содержащие вопросы как теоретического, так и практического характера, которые помогут студентам изучить дисциплину в самостоятельном режиме работы.

По программе курса выполняются проверочные работы по каждому разделу. Проверочные работы могут быть проведены в форме теста (в соответствии с рабочей программой дисциплины). После изучения всей дисциплины студенты защищают выполненное расчетно-графическое задание. Из тестовых вопросов составляется итоговый тест по всему курсу «Дискретная математика».

ВВЕДЕНИЕ

Дискретная математика, или дискретный анализ, – область математики, которая занимается исследованием структур и задач на конечных множествах. Поэтому в качестве синонима иногда используется термин «конечная математика». Можно считать общепринятым деление математики на непрерывную и дискретную. Последняя представляет собой важное направление, имеющее характерные для него предмет исследований, методы и задачи. Специфика задач дискретной математики в первую очередь предполагает отказ от основных понятий классической математики – предела и непрерывности.

Дискретная и непрерывная математика взаимно дополняют друг друга. Понятия и методы одной часто используются в другой. Один и тот же объект может рассматриваться с двух точек зрения и в зависимости от этого выбирается непрерывная или дискретная математика.

Графические представления в широком смысле – любые наглядные отображения исследуемой системы, процесса, явления на плоскости. В последние несколько лет теория графов стала важнейшим математическим инструментом, широко используемым во многих областях науки, начиная с исследования операций и лингвистики и заканчивая химией и генетикой. В то же самое время она выросла в самостоятельную математическую дисциплину. Для ее понимания требуется только знание элементарной теории множеств и теории матриц.

На протяжении последних десятилетий велись и ведутся интенсивные работы по созданию и использованию различных систем и устройств для переработки дискретной информации. *Преобразователи дискретной информации* широко используются в качестве различного рода технических автоматов, вычислительных устройств и их функциональных блоков, устройств управления роботами, управляющих объектами по заданному алгоритму. Широкий класс таких преобразователей объединяется под общим названием – ***автоматы***.

Теория автоматов находит применение как в математике, так и в решении практических задач. Применение методов и понятий теории автоматов к изучению формальных и естественных языков привело к возникновению математической лингвистики (математическая лингвистика – математическая дисциплина, предметом которой является разработка формального аппарата для описания строения естественных и некоторых искусственных языков). Понятие автомата может служить модельным объектом в самых разнообразных задачах, благодаря чему возможно применение теории автоматов в различных научных и прикладных исследованиях.

1. ТЕОРИЯ ГРАФОВ

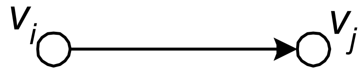
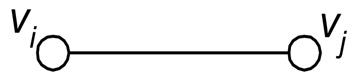
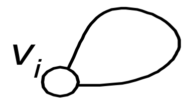
1.1. Основные определения

*Определение 1.1. Пара (V(G), E(G)) называется **простым графом**, если V(G) – непустое конечное множество элементов, называемых **вершинами** (или узлами, или точками), а E(G) – конечное множество неупорядоченных пар различных элементов из V(G), называемых **ребрами** (или линиями). В простом графе данную пару вершин может соединять не более чем одно ребро* (рис. 1.1).

Между элементами множества вершин и множества ребер определено отношение *инцидентности*. Говорят, что ребро e инцидентно вершинам v_1, v_2, если оно соединяет эти вершины, и наоборот, каждая из вершин v_1, v_2 инцидентна ребру e.

Рассмотрим графическое представление графов (табл. 1.1).

Таблица 1.1

Элементы графов	Геометрические элементы	Наименование элемента
$x \in X$ – *вершина*		Точка в пространстве
$(v_i, v_j) \in V \wedge (v_i, v_j) \notin V$		Направленный отрезок, ориентированное ребро
$(v_i, v_j) \in V \wedge (v_i, v_j) \in V$		Отрезок, неориентированное ребро
$(v_i, v_i) \in V$		Замкнутый отрезок, петля

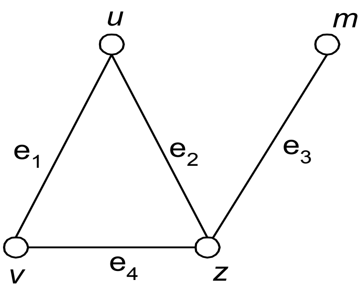


Рис. 1.1

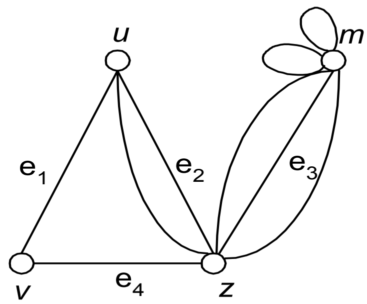


Рис. 1.2

Многие результаты, полученные для простых графов, без труда можно перенести на более общие объекты, в которых две вершины могут быть соединены более чем одним ребром. Кроме того, часто бывает удобно снять ограничение, состоящее в том, что ребро должно соединять две различные вершины, и допустить существование петель. Получающийся при этом объект, в котором могут быть и кратные ребра, и петли, называется ***графом*** (***псевдографом***) (рис. 1.2). Псевдограф без петель называется ***мультиграфом***.

Определение 1.2. *Если ребро задаётся упорядоченной парой вершин, то оно является **ориентированным**. Если каждое ребро графа ориентированное, то граф называется **ориентированным**, или **орграфом**.*

Иногда удобно преобразовать неориентированный граф в ориентированный – заменой одного неориентированного ребра парой ребер с противоположной ориентацией.

Приведем ряд понятий и определений для ориентированных и неориентированных графов. Там, где это специально не оговорено, те же понятия и определения переносятся без изменений на ориентированные и неориентированные псевдографы.

1.1.1. Смежность, инцидентность, степени

Определение 1.3. *Если $e = \{v, w\}$ – ребро графа, то вершины v, w называются* ***концами ребра e****; в этом случае также говорят, что* ***ребро*** *e* ***соединяет вершины*** *v, w.*

Определение 1.4. *Если $e = \{v, w\}$ – дуга орграфа, то вершина v называется* ***началом****, а вершина w –* ***концом дуги*** *e; в этом случае также говорят, что дуга e исходит из вершины v и заходит в вершину w.*

Между элементами множества вершин и множества ребер определено отношение *инцидентности*. Говорят, что ребро e инцидентно вершинам v, w, если оно соединяет эти вершины, и наоборот, каждая из вершин v, w инцидентна ребру e.

Определение 1.5. *Две вершины называются* ***смежными****, если существует ребро, концами которого они являются. Два ребра называются смежными, если они имеют общую вершину.*

Определение 1.6. ***Степенью*** *вершины v графа G называется число $\delta(v)$ ребер графа, которым инцидентна эта вершина.*

Определение 1.7. *Вершина, локальная степень которой равна 0, называется* ***изолированной****; вершина, степень которой равна 1, –* ***висячей****.*

Замечание. *В случае неориентированного псевдографа обычно считается, что вклад каждой петли, инцидентной некоторой вершине v, равен 2 (тогда как вклад любого другого ребра, инцидентного вершине v, равен 1).*

Определение 1.8. ***Полустепенью исхода (захода)*** *вершины v орграфа D называется число $\delta^+(v)$ ($\delta^-(v)$) дуг орграфа D, исходящих из вершины v (заходящих в вершину v).*

Замечание. *В случае ориентированного псевдографа вклад каждой петли, инцидентной некоторой вершине v, равен 1, как в $\delta^+(v)$, так и в $\delta^-(v)$.*

Количество вершин и ребер в графе G обозначим соответственно через $n(G)$ и $m(G)$, а количество вершин и дуг в орграфе D – через $n(D)$ и $m(D)$.

Утверждения.

1) Для любого псевдографа G выполняется равенство $\sum\limits_{v\in V}\delta(v) = 2m(G)$.

2) Для любого ориентированного псевдографа D выполняется равенство $\sum\limits_{v\in V}\delta^+(v) = \sum\limits_{v\in V}\delta^-(v) = m(G)$.

Пример 1.1

Найти локальные степени графа (рис. 1.3) и орграфа (рис. 1.4).

Решение.

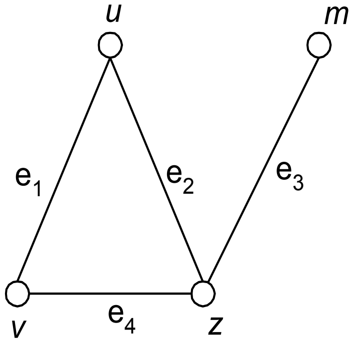


Рис. 1.3

$\delta(u) = 2;$
$\delta(v) = 2;$
$\delta(z) = 3;$
$\delta(m) = 1.$

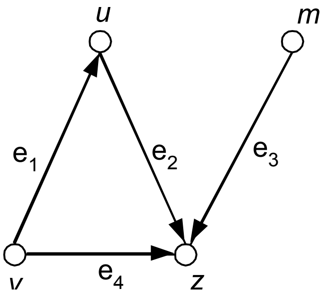


Рис. 1.4

$\delta^+(u) = 1;$ $\delta^-(u) = 1;$
$\delta^+(v) = 2;$ $\delta^-(v) = 0;$
$\delta^+(z) = 0;$ $\delta^-(z) = 3;$
$\delta^+(m) = 1;$ $\delta^-(m) = 0.$

1.1.2. Изоморфизм, гомеоморфизм

Определение 1.9. Два графа G_1 и G_2 называются ***изоморфными****, если существует взаимно однозначное соответствие между множествами их вершин, обладающее тем свойством, что число ребер, соединяющих любые две вершины в G_1, равно число ребер, соединяющих соответствующие вершины в G_2.*

Из определения следует, что изоморфные графы можно одинаково изображать графически и отличаться они будут только метками вершин.

Изоморфизм графов есть отношение эквивалентности.

На рис. 1.5 изображены три изоморфных графа, на рис. 1.6 – два неизоморфных графа.

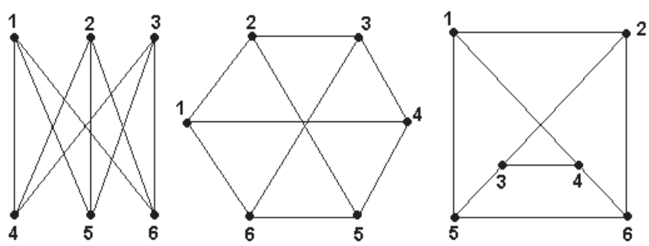


Рис. 1.5

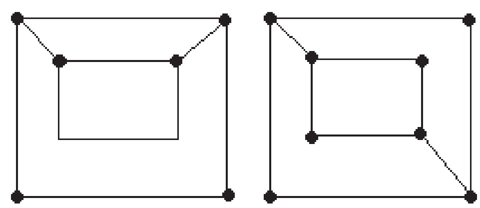


Рис. 1.6

Графы рассматриваются с точностью до изоморфизма, то есть рассматриваются классы эквивалентности по отношению изоморфизма.

Пример 1.2

С точностью до изоморфизма существует ровно четыре простых графа с тремя вершинами и одиннадцать с четырьмя вершинами. Постройте эти графы.

Решение.

1) Построим четыре простых графа с тремя вершинами с точностью до изоморфизма (рис. 1.7).

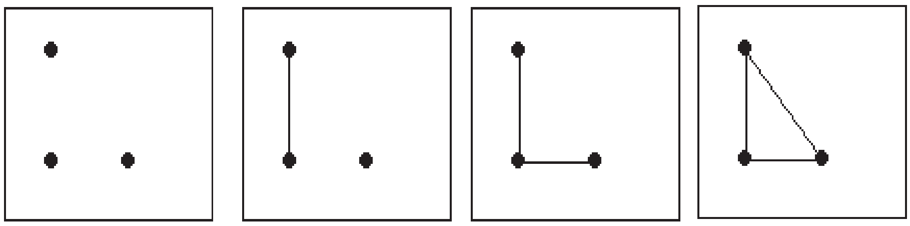


Рис. 1.7

2) Построим одиннадцать простых графов с четырьмя вершинами с точностью до изоморфизма (рис. 1.8):

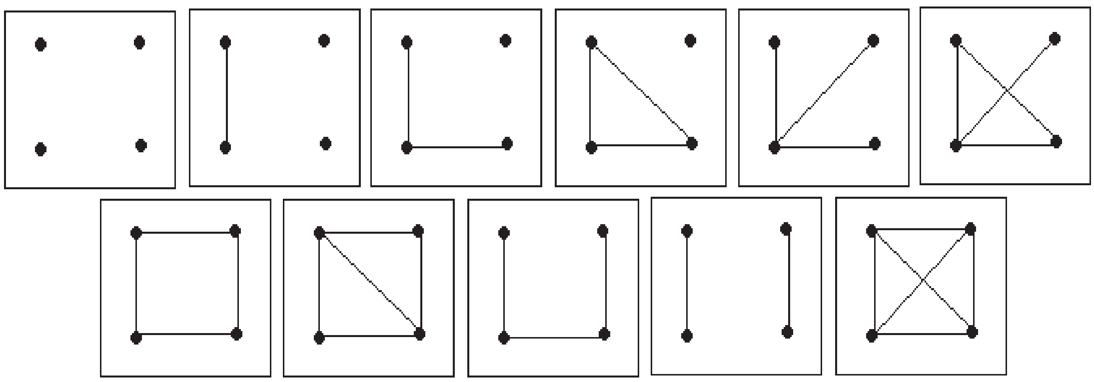


Рис. 1.8

Определение 1.10. ***Операция подразбиения (измельчения) дуги*** *(u, v) в орграфе $D = (V, E)$ состоит в удалении из E дуги (u, v), добавлении к V новой вершины w и добавлении к $E \mid \{(u, v)\}$ двух дуг (u, v), (w, v). Аналогично определяется операция подразбиения ребра в графах.*

Определение 1.11. Орграф D_1 называется ***подразбиением*** *орграфа D_2, если орграф D_1 можно получить из D_2 путем последовательного применения операции подразбиения дуг. Аналогично определяется подразбиение графа.*

Определение 1.12. Орграфы D_1, D_2 называются ***гомеоморфными****, если существуют их подразбиения, являющиеся изоморфными.*

1.1.3. Матричное задание графов. Матрицы смежности, инцидентности

Пусть $D = (V, X)$ – ***орграф***, где $V = \{v_1, v_2, \ldots, v_n\}$, $X = \{x_1, x_2, \ldots, x_m\}$.

Определение 1.13. ***Матрицей смежности*** *орграфа D называется квадратная матрица $A(D) = [a_{ij}]$ порядка n, у которой*

$$a_{ij} = \begin{cases} 1, \text{если } (v_i, v_j) \in X; \\ 0, \text{если } (v_i, v_j) \notin X. \end{cases}$$

Определение 1.14. ***Матрицей инцидентности*** *орграфа D называется $(n \times m)$ –матрица $B(D) = [b_{ij}]$, у которой*

$$b_{ij} = \begin{cases} 1, \text{ если вершина } v_i \text{ является концом дуги } x_j; \\ -1, \text{ если вершина } v_i \text{ является началом дуги } x_j; \\ 0, \text{ если вершина } v_i \text{ не инцидентна дуге } x_j. \end{cases}$$

Введем также матрицы смежности и инцидентности для неориентированных графов. Пусть $G = (V, X)$ – граф, где $V = \{v_1, v_2, \ldots, v_n\}$, $X = \{x_1, x_2, \ldots, x_m\}$.

Определение 1.15. ***Матрицей смежности*** *графа G называется квадратная матрица $A(G)=[a_{ij}]$ порядка n, у которой*

$$a_{ij} = \begin{cases} 1, \text{если } (v_i, v_j) \in X; \\ 0, \text{если } (v_i, v_j) \notin X. \end{cases}$$

Определение 1.16. ***Матрицей инцидентности*** *графа G называется $(n \times m)$ –матрица $B(G) = [b_{ij}]$, у которой*

$$b_{ij} = \begin{cases} 1, \text{ если вершина } v_i \text{ инцидентна ребру } x_j; \\ \\ 0, \text{ если вершина } v_i \text{ не инцидентна ребру } x_j. \end{cases}$$

С помощью введенных матриц удобно задавать графы для обработки на ЭВМ. Используя матрицу смежности, легко определить локальные степени вершин графа: сумма элементов матрицы по строке равна локальной степени соответствующей вершины. Для орграфов по строке определяются полустепени исхода, по столбцам – полустепени захода.

Пример 1.3

Построить матрицы смежности и инцидентности для графа $G = (V, X)$ (рис. 1.9).

Решение.

Матрица смежности имеет вид

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

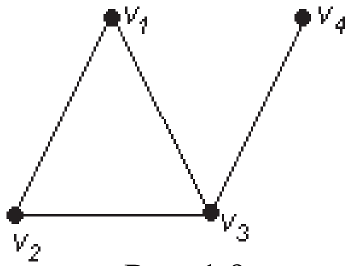


Рис. 1.9

Поскольку граф не имеет петель, то на главной диагонали стоят все нули. Для любого графа матрица смежности симметрична относительно главной диагонали.

Для того чтобы построить матрицу инцидентности, необходимо пронумеровать ребра графа (рис. 1.10). Матрица инцидентности имеет вид

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

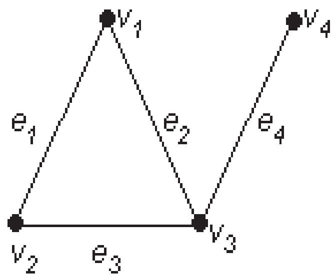


Рис. 1.10

Напомним, что в строках указываются вершины, в столбцах – ребра. Матрица инцидентности может быть как квадратной, так и прямоугольной.

Пример 1.4

Построить матрицы смежности и инцидентности для орграфа $D = (V, X)$ (рис. 1.11).

Решение.

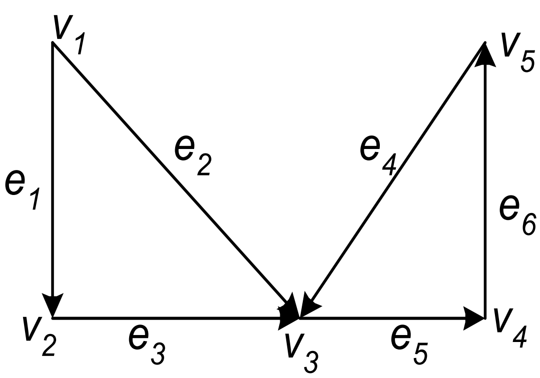


Рис. 1.11

Матрица смежности имеет вид

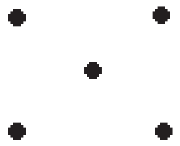
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Матрица инцидентности имеет вид

$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

1.1.4. Примеры графов. Операции над графами

Рассмотрим некоторые важные типы графов.



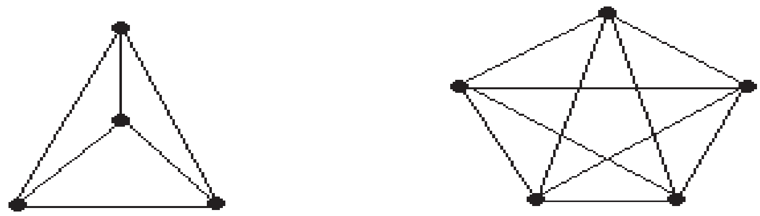
N_5

Рис. 1.12

Определение 1.17. *Граф, у которого множество ребер пусто, называется* ***вполне несвязным*** *(или* ***пустым****) графом. Вполне несвязный граф обозначают* N_n (рис. 1.12).

Заметим, что у вполне несвязного графа все вершины изолированы.

Определение 1.18. *Простой граф, в котором любые две вершины смежны, называется* ***полным***. *Полный граф обозначают* K_n.(рис. 1.13).



K_4 K_5

Рис. 1.13

Заметим, что для полного графа выполняется равенство $m = \dfrac{n(n-1)}{2}$, где m – число ребер, n – число вершин графа.

Определение 1.19. *Граф, у которого все вершины имеют одну и ту же локальную степень n, называется* ***регулярным*** *(или* ***однородным****) степени n.*

Регулярные графы степени 3 называются *кубическими* (или *трехвалентными*).

Известным примером кубического графа является граф Петерсена (рис. 1.14).

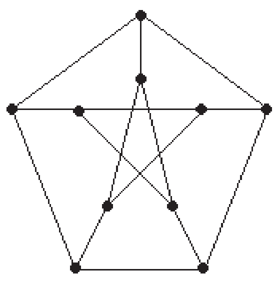


Рис. 1.14

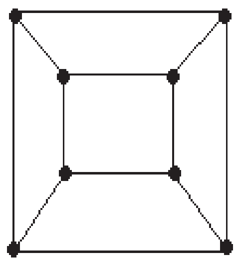


Рис. 1.15

Среди регулярных графов особенно интересны так называемые *платоновы графы* – графы, образованные вершинами и ребрами пяти правильных многогранников – Платоновых тел: тетраэдра, куба, октаэдра, додекаэдра и икосаэдра.

На рис. 1.15 приведен граф, соответствующий кубу.

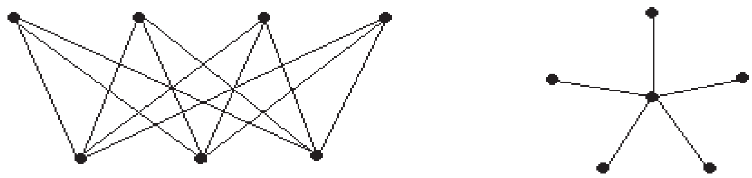
Определение 1.20. *Допустим, что множество вершин графа G можно разбить на два непересекающихся подмножества V_1 и V_2 так, что каждое ребро в G соединяет какую-нибудь вершину из V_1 с какой-нибудь вершиной из V_2, тогда данный граф называется* ***двудольным****.*

Двудольный граф можно определить и по-другому – в терминах раскраски его вершин двумя цветами, скажем красным и синим. При этом граф называется двудольным, если каждую его вершину можно окрасить красным или синим цветом так, чтобы каждое ребро имело один конец красный, а другой – синий.

Определение 1.21. *Если в двудольном графе каждая вершина из V_1 соединена с каждой вершиной из V_2, то граф называется* ***полным двудольным****. Обозначение $K_{m.\,n}$.*

Заметим, что граф $K_{m.\,n}$ имеет ровно $m + n$ вершин и $m*n$ ребер.

На рис. 1.16 изображены двудольные графы.



$K_{4,\,3}$ $\qquad\qquad$ $K_{1,\,5}$

Рис. 1.16

Определение 1.22. ***Объединением*** *графов $G_1 = (V_1, X_1)$, $G_2 = (V_2, X_2)$ называется граф $G_1 \cup G_2 = (V_1 \cup V_2,\ X_1 \cup X_2)$.*

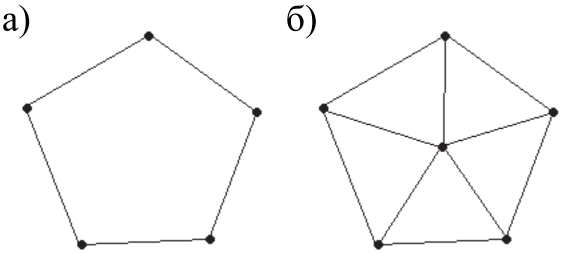
Определение 1.23. ***Пересечением*** *графов $G_1 = (V_1, X_1)$, $G_2 = (V_2, X_2)$, где $V_1 \cap V_2 \neq \varnothing$, называется граф $G_1 \cap G_2 = (V_1 \cap V_2,\ X_1 \cap X_2)$.*

Определение 1.24. ***Соединением*** *графов G_1 и G_2 является новый граф, у которого $V = V_1 \cup V_2$, а множеством ребер являются все ребра первого и второго графа и ребра, соединяющие между собой каждую вершину первого графа с первой вершиной второго графа.*

Определение 1.25. *Граф называется* ***связным****, если его нельзя представить в виде объединения двух графов, и несвязным в противном случае.*

Очевидно, что всякий несвязный граф можно представить в виде объединения конечного числа связных графов – каждый из таких связных графов называется *компонентой связности* графа.

а) б)



C_5 W_6

Рис. 1.17

Определение 1.26. Связный регулярный граф степени 2 называется ***циклическим*** *графом. Обозначается он* C_n (рис. 1.17, *а*).

Определение 1.27. Соединение графов N_1 *и* C_{n-1} ($n \geq 3$) *называется* ***колесом*** *с n вершинами. Обозначается* W_n (рис. 1.17, *б*).

Определение 1.28. ***Дополнением*** *простого графа G называется простой граф с множеством вершин V(G), в котором две вершины смежны тогда и только тогда, когда они не смежны в исходном графе. Обозначается* $\overline{G}$.

Другими словами, дополнением графа является граф, содержащий все вершины исходного графа и только те ребра, которых не хватает исходному графу для того, чтобы он стал полным.

Определение 1.29. ***Подграфом*** *графа G называется граф, все вершины и ребра которого содержатся среди вершин и ребер графа G. Подграф называется* ***собственным****, если он отличен от самого графа.*

1.1.5. Маршруты, цепи, циклы

Введем понятие маршрута для графа $G = (V, E)$ (и соответственно понятие пути для орграфа $D = (V, E)$).

Определение 1.30. ***Маршрутом*** *в данном графе G называется конечная последовательность ребер вида* $\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{m-1}, v_m\}$ (*обозначаемая также* $v_0 \to v_1 \to v_2 \to \ldots \to v_m$).

Каждому маршруту соответствует последовательность вершин $v_0v_1v_2\ldots v_m$; v_0 – начальная вершина, а v_m – конечная вершина маршрута. Одна и та же вершина может одновременно оказаться начальной, конечной и внутренней. Таким образом, мы будем говорить о маршруте из v_0 в v_m.

Определение 1.31. ***Длиной маршрута*** *называется число ребер в нем.*

Определение 1.32. Маршрут называется ***цепью****, если все его ребра различны, и* ***простой цепью****, если все вершины тоже различны (кроме, может быть, начальной и конечной вершин).*

Определение 1.33. Цепь (или простая цепь) ***замкнута****, если начальная и конечная вершины совпадают.*

Определение 1.34. Замкнутая простая цепь, содержащая, по крайней мере, одно ребро, называется ***циклом****.*

Определение 1.35. ***Обхватом*** *графа называется длина его кратчайшего цикла.*

1.1.6. Связность. Компоненты связности

Определение 1.36. ***Вершина*** *w орграфа D (графа G)* ***достижима*** *из вершины v, если либо v = w, либо существует путь из v в w(маршрут, соединяющий v, w).*

Дадим более удобное определение связных графов.

Определение 1.37. Граф называется ***связным****, если для любых двух его вершин v, w существует простая цепь из v в w.*

Определение 1.38. Граф (орграф) называется ***связным (сильно связным)****, если для любых двух его вершин v, w существует маршрут (путь), соединяющий v, w (из v в w).*

Определение 1.39. Орграф называется ***односторонне связным****, если для любых его двух вершин, по крайней мере, одна достижима из другой.*

Определение 1.40. Если граф не является связным, то он называется ***несвязным****.*

Определение 1.41. ***Компонентой связности*** *графа называется его связный подграф, не являющийся собственным подграфом никакого другого связного подграфа.*

В дальнейшем количество компонент связности графа будем обозначать k.

Например, граф, приведенный на рис. 1.18, *а* не является связным: $k = 3$, а граф, приведенный на рис. 1.18, *б*, является связным: $k = 0$.

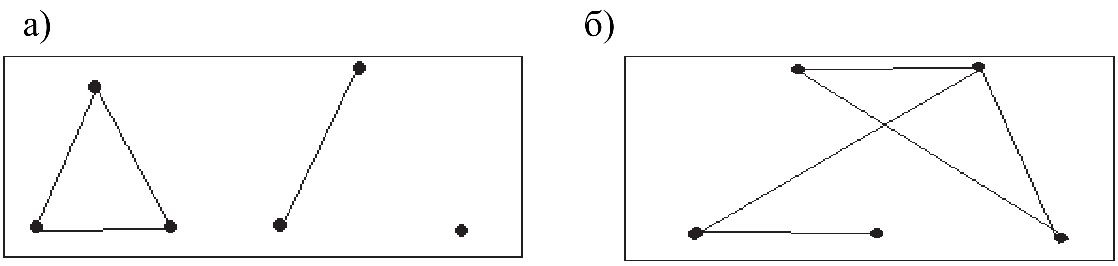


Рис. 1.18

Теорема. *Пусть G – простой граф с n вершинами и k компонентами. Тогда число m его ребер удовлетворяет неравенствам*

$$n-k \le m \le \frac{(n-k)(n-k+1)}{2}.$$

Следствие. *Любой простой граф с n вершинами и более чем* $(m - 1)(m - 2)/2$ *ребрами связен.*

При исследовании графов возникает вопрос: насколько сильно связен связный граф? Этот вопрос можно сформулировать и так: сколько ребер нужно удалить из графа, чтобы он перестал быть связным? Под операцией удаления вершин из графа будем понимать операцию, заключающуюся в удалении некоторой вершины вместе с инцидентными ей ребрами.

Определение 1.42. *Вершина графа, удаление которой увеличивает число компонент связности, называется* ***разделяющейся***.

Определение 1.43. ***Разделяющим множеством*** *связного графа G называется такое множество его ребер, удаление которого приводит к несвязному графу.*

Определение 1.44. ***Разрезом*** *называется такое разделяющее множество, никакое собственное подмножество которого не является разделяющим.*

Определение 1.45. *Разрез, состоящий из одного ребра, называется* ***мостом*** *(перешейком).*

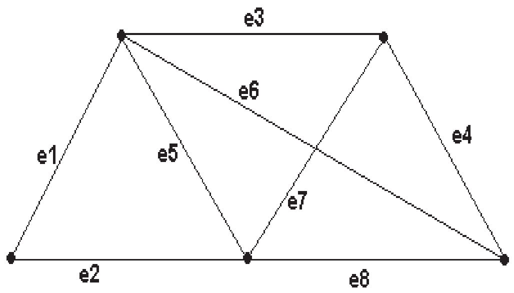


Рис. 1.19

Для графа, изображенного на рис. 1.19, каждое из множеств $\{e_1, e_2, e_5\}$ и $\{e_3, e_6, e_7, e_8\}$ является разделяющим.

Разрезом является множество ребер $\{e_1, e_2\}$.

В графе возможно выделить несколько разделяющих множеств и разрезов.

1.2. Задачи поиска маршрутов (путей) в графе (орграфе)

1.2.1. Поиск путей (маршрутов) с минимальным числом дуг (ребер)

При решении некоторых прикладных задач нередко возникает необходимость найти маршрут, соединяющий заданные вершины в графе G. Данная задача решается при использовании алгоритма Тэрри.

Рассмотрим более сложную задачу: нахождение пути (маршрута) с минимальным числом дуг (ребер).

Определение 1.46. Путь в орграфе D из вершины v в вершину w, где $v \neq w$, называется ***кратчайшим****, если он имеет минимальную длину среди всех путей орграфа D из v и w. Аналогично определяется и кратчайший маршрут в графе G.*

Утверждение. Любой кратчайший путь (маршрут) является простой цепью.

Рассмотрим задачу поиска минимального пути (маршрута). Введем некоторые обозначения. Пусть $D = (V, X)$ – орграф, $v \in V$, $V_1 \subset V$. Обозначим $D(v) = \{w \in V \mid (v, w) \in X\}$ – ***образ вершины*** v; $D^{-1}(v) = \{w \in V \mid (w,v) \in X\}$ – ***прообраз вершины*** v; $D(V_1) = \bigcup\limits_{v \in V_1} D(v)$ – образ множества вершин V_1; $D^{-1}(V_1) = \bigcup\limits_{v \in V_1} D^{-1}(v)$ – прообраз множества вершин V_1. Пусть $G = (V, X)$ – граф, $v \in V$, $V_1 \subset V$. Обозначим $G(v) = \{w \in V | \{v, w\} \in X\}$ – образ вершины v; $G(V_1) = \bigcup\limits_{v \in V_1} G(v)$ – образ множества вершин V_1.

Пусть $D = (V, X)$ – орграф с $n \geq 2$ вершинами и v, w – заданные вершины из V, где $v \neq w$. Опишем алгоритм поиска кратчайшего пути из v в w в орграфе D.

Алгоритм «Фронта волны»

Шаг 1. Помечаем вершину v индексом 0. Затем помечаем вершины, принадлежащие образу вершины v, индексом 1. Множество вершин с индексом 1 обозначаем $FW_1(v)$. Полагаем $k = 1$.

Шаг 2. Если $FW_k(v) = \varnothing$ или выполняется $k = n - 1$, $w \notin FW_k(v)$, то вершина w не достижима из v, и работа алгоритма на этом заканчивается. В противном случае переходим к шагу 3.

Шаг 3. Если $w \notin FW_k(v)$, то переходим к шагу 4. В противном случае существует путь из v в w длины k, и этот путь является кратчайшим. Последовательность вершин

$$w_1 w_2 \ldots w_{k-1} w,$$

где

$$w_{k-1} \in FW_{k-1}(v) \cap D^{-1}(w);$$

$$w_{k-2} \in FW_{k-2}(v) \cap D^{-1}(w_{k-1});$$

....

$$w_1 \in FW_1(v) \cap D^{-1}(w_2),$$

и есть искомый кратчайший путь из v и w. На этом работа алгоритма заканчивается.

Шаг 4. Помечаем индексом $k+1$ все непомеченные вершины, которые принадлежат образу множества вершин с индексом k. Множество вершин с индексом $k+1$ обозначаем $FW_{k+1}(v)$. Присваиваем $k:=k+1$ и переходим к шагу 2.

Замечания.

1) Множество $FW_k(v)$ обычно называют фронтом волны k-го уровня.

2) Вершины $w_1 w_2 ... w_{k-1}$, вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных кратчайших путей из v и w.

Пример 1.5

Определить кратчайший путь из v_1 и v_6 в орграфе D, заданном матрицей смежности (табл. 1.2).

Таблица 1.2

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	0	0	1	1	0
v_2	1	0	0	1	1	1
v_3	1	1	0	1	1	1
v_4	0	1	1	0	1	0
v_5	1	1	1	1	0	0
v_6	1	1	1	1	1	0

Решение.

Согласно алгоритму «Фронта волны», последовательно определяем

$FW_1(v_1) = \{v_4, v_5\}$;

$FW_2(v_1) = D(FW_1(v_1))/\{v_1, v_4, v_5\} = = \{v_2, v_3\}$;

$FW_3(v_1) = D(FW_2(v_1))/\{v_1, v_2, v_3, v_4, v_5\} = \{v_6\}$.

Таким образом, $v_6 \in FW_3(v_1)$, а значит, существует путь из v_1 в v_6 длины 3, и этот путь является кратчайшим. Найдем теперь кратчайший путь из v_1 в v_6. Определим множество

$$FW_2(v_1) \cap D^{-1}(v_6) = \{v_2, v_3\} \cap \{v_2, v_3\} = \{v_2, v_3\}.$$

Выберем любую вершину из найденного множества, например вершину v_3. Определим далее множество

$$FW_1(v_1) \cap D^{-1}(v_3) = \{v_4, v_5\} \cap \{v_4, v_5, v_6\} = \{v_4, v_5\}.$$

Выберем любую вершину из найденного множества, например вершину v_5. Тогда $v_1 v_5 v_3 v_6$ – искомый кратчайший путь из v_1 в v_6 (длины 3) в орграфе D.

1.2.2. Расстояния в графе. Диаметр, центр, радиус графа

Утверждение. *Если для двух вершин существует маршрут, связывающий их, то обязательно найдется минимальный маршрут, соединяющий эти вершины. Обозначим длину этого маршрута через $d(v, w)$.*

Определение 1.47. *Величину $d(v, w)$ (конечную или бесконечную) будем называть **расстоянием между вершинами v, w**. Это расстояние удовлетворяет аксиомам метрики:*

1) $d(v, w) \geq 0$, причем $d(v, w) = 0$ тогда и только тогда, когда $v = w$;
2) $d(v, w) = d(w, v)$;
3) $d(v, w) \leq d(v, u) + d(u, w)$.

Определение 1.48. ***Диаметром*** *связного графа называется максимально возможное расстояние между двумя его вершинами.*

Определение 1.49. ***Центром*** *графа называется такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных; это расстояние называется* ***радиусом*** *графа.*

Пример 1.6

Для графа G, изображенного на рис. 1.20, найти радиус, диаметр и центры.

Решение.

Чтобы определить центры, радиус, диаметр графа G, найдем матрицу $D(G)$ расстояний между вершинами графа, элементами d_{ij} которой будут расстояния между вершинами v_i и v_j. Для этого воспользуемся графическим представлением графа.

$$D(G) = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 0 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 0 & 2 \\ 3 & 2 & 1 & 2 & 0 \end{bmatrix}.$$

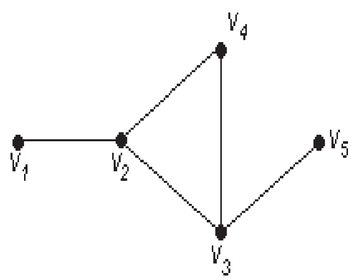


Рис. 1.20

Заметим, что матрица $D(G)$ симметрична относительно главной диагонали.

С помощью полученной матрицы для каждой вершины графа G определим наибольшее удаление из выражения: $r(v_i) = \max_j d(v_i, v_j)$ для $i, j = 1, 2, \ldots, 5$. В результате получим: $r(v_1) = 3$, $r(v_2) = 2$, $r(v_3) = 2$, $r(v_4) = 2$, $r(v_5) = 3$. Минимальное из полученных чисел является радиусом графа G, максимальное – диаметром графа G. Значит, $R(G) = 2$ и $D(G) = 3$, центрами являются вершины v_2, v_3, v_4.

1.2.3. Нахождение минимального пути в нагруженном графе

Определение 1.50. Назовём орграф $D=(V,X)$ ***нагруженным****, если на множестве дуг X определена некоторая функция $l: X \to R$, которую часто называют* ***весовой*** *функцией.*

Тем самым в нагруженном орграфе *D* каждой дуге $x \in X$ поставлено в соответствие некоторое действительное число $l(x)$. Значение $l(x)$ будем называть *длиной дуги* x.

Для любого пути π нагруженного орграфа *D* обозначим через $l(\pi)$ сумму длин входящих в π дуг, при этом каждая дуга учитывается столько раз, сколько она входит в путь. Величину $l(\pi)$ будем называть *длиной пути* π в нагруженном орграфе *D*. Ранее так называлось количество дуг в пути π. В связи с этим заметим, что если длины дуг выбраны равными 1, то $l(\pi)$ выражает введенную ранее длину пути π в ненагруженном орграфе. Следовательно, любой ненагруженный орграф можно считать нагруженным с длинами дуг, равными 1. Аналогично определяется и нагруженный граф, а также длина маршрута в нем.

Определение 1.51. Путь в нагруженном орграфе D из вершины υ в вершину w, где $\upsilon \neq w$, называется ***минимальным****, если он имеет минимальную длину среди всех путей орграфа D из υ в w. Аналогично определяется и минимальный маршрут в нагруженном графе G.*

Рассмотрим теперь задачу поиска минимальных путей (маршрутов) в нагруженном орграфе (графе). При этом для определенности рассуждения будем проводить для орграфа (для графа они аналогичны).

Пусть $D=(V,X)$ – нагруженный орграф, $V=\{\upsilon_1,...,\upsilon_n\}$, $n \geq 2$. Введем величины $\lambda_i^{(k)}$, где $i=1, ..., n$, $k=1, 2, ... n$-1. Для каждых фиксированных i и k величина $\lambda_i^{(k)}$ равна длине минимального пути среди путей из υ_1 в υ_i, содержащих не более k дуг; если же таких путей нет, то $\lambda_i^{(k)} = \infty$. Кроме того, если произвольную вершину $\upsilon \in V$ считать путем из υ в υ нулевой длины, то величины $\lambda_i^{(k)}$ можно ввести также и для $k=0$, при этом

$$\lambda_1^{(0)} = 0, \lambda_i^{(0)} = \infty, \ i = 2, ..., n. \tag{1.1}$$

Введем также в рассмотрение квадратную матрицу $C(D)=\lfloor c_{ij} \rfloor$ порядка *n* с элементами

$$c_{ij} = \begin{cases} l(\upsilon_i, \upsilon_j), \text{ если } (\upsilon_i, \upsilon_j) \in X; \\ \infty, \text{ если } (\upsilon_i, \upsilon_j) \overline{\in} X, \end{cases}$$

которую будем называть *матрицей длин дуг* нагруженного орграфа *D*.

Следующее утверждение дает простые формулы для вычисления величин $\lambda_i^{(k)}$.

Утверждение. При $i = 2,...,n,\ k \geq 0$ выполняется равенство

$$\lambda_i^{(k+1)} = \min_{1 \leq j \leq n} \left\{ \lambda_j^{(k)} + c_{ji} \right\}, \tag{1.2}$$

а при $i = 1,\ k \geq 0$ справедливо равенство

$$\lambda_1^{(k+1)} = \min_{1 \leq j \leq n} \left\{ \lambda_j^{(k)} + c_{j1} \right\}. \tag{1.3}$$

Используя данное утверждение, нетрудно описать алгоритм нахождения таблицы значений величин $\lambda_i^{(k)}$ (будем записывать её в виде матрицы, где i – номер строки, $k+1$ – номер столбца). Действительно, используя рекуррентные соотношения (1.2), (1.3) и исходя из (1.1), последовательно определяем набор величин $\lambda_1^{(k)}$, …, $\lambda_n^{(h)}$ (($k+1$)-й столбец матрицы), начиная с $k+1$, а затем шаг за шагом увеличиваем значение k до любой необходимой величины.

Алгоритм Форда–Беллмана нахождения минимального пути в нагруженном орграфе D из υ_1 в υ_{i_1} $(i_1 \neq 1)$

Шаг 1. Пусть мы уже составили таблицу величин $\lambda_i^{(k)}, i = 1, 2, ..., n, k = 0, 1, ..., n-1$. Если $\lambda_{i_1}^{(n-1)} = \infty$ не достижима из υ_1 (предполагаем, что все величины $l(x),\ \ x \in X$, конечны). В этом случае работа алгоритма заканчивается.

Шаг 2. Пусть $\lambda_{i_1}^{(n-1)} < \infty$. Тогда число $\lambda_{i_1}^{(n-1)}$ выражает длины любого минимального пути из υ_1 в υ_{i_1} в нагруженном орграфе *D*. Определим минимальное число $k_1 \geq 1$, при котором выполняется равенство $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)}$. По определению чисел $\lambda_i^{(k)}$ получим, что k_1 – минимальное число дуг в пути среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе *D*.

Шаг 3. Последовательно определяем номера $i_2, ..., i_{k_1+1}$ такие что

$$\begin{aligned} &\lambda_{i_2}^{(k_1-1)} + c_{i_2, i_1} = \lambda_{i_1}^{(k_1)}; \\ &\lambda_{i_3}^{(k_1-2)} + c_{i_3, i_2} = \lambda_{i_2}^{(k_1-1)}; \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &\lambda_{i_{k_1+1}}^{(0)} + c_{i_{k_1+1}, i_{k_1}} = \lambda_{i_{k_1}}^{(1)} \end{aligned} \tag{1.4}$$

Из (1.4) с учётом того, что $\lambda_{i_1}^{(k_1)} = \lambda_{i_1}^{(n-1)} < \infty$, имеем $c_{i_2,i_1} < \infty, ..., c_{i_{k_1+1},i_{k_1}} < \infty$, $\lambda_{i_{k_1+1}}^{(0)} < \infty$, откуда, используя (1.1), получаем:

$$\begin{gathered}(\upsilon_{i_2},\upsilon_{i_1}),...,(\upsilon_{i_{k_1+1}},\upsilon_{i_{k_1}}) \in X, \;\; l(\upsilon_{i_2},\upsilon_{i_1}) = c_{i_2,i_1},...,l\;\;(\upsilon_{i_{k_1+1}},\upsilon_{i_{k_1}}) = c_{i_{k_1+1},i_{k_1}};\\ \lambda_{i_{k_1+1}}^{(0)} = 0, i_{k_1}+1 = 1, \upsilon_{i_{k_1+1}} = \upsilon_1.\end{gathered} \tag{1.5}$$

Складывая равенства (1.4) и учитывая (1.5), имеем

$$l(\upsilon_1\upsilon_{k_1} \ldots \upsilon_{i_2}\upsilon_{i_1}) = \lambda_{i_1}^{(k_1)},$$

т.е. $\upsilon_1\upsilon_{k_1} \ldots \upsilon_{i_2}\upsilon_{i_1}$ – искомый минимальный путь из υ_1 в υ_{i_1} в нагруженном орграфе D. Заметим, что в этом пути ровно k_1 дуг. Следовательно, мы определили путь с минимальным числом дуг среди всех минимальных путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

Замечания. 1) Номера $i_2, i_3, ..., i_{k_1}$, удовлетворяющие (1.4) вообще говоря, могут быть выделены неоднозначно. Эта неоднозначность соответствует случаям, когда существует несколько различных путей из υ_1 в υ_{i_1} с минимальным числом дуг среди минимальных, путей из υ_1 в υ_{i_1} в нагруженном орграфе D.

2) Алгоритм можно модифицировать, с тем, чтобы определить минимальный путь из υ_1 в заданную вершину υ_{i_1} среди путей из υ_1 в υ_{i_1}, содержащих не более k_0 дуг, где k_0 – заданное число, $k_0 \geq 1$. Для этого в алгоритме вместо $\lambda_{i_1}^{(n-1)}$ следует воспользоваться $\lambda_{i_1}^{(k_0)}$.

Пример 1.7

Определить минимальный путь из v_1 в v_6 в нагруженном орграфе D, изображенном на рис. 1.21.

Решение.

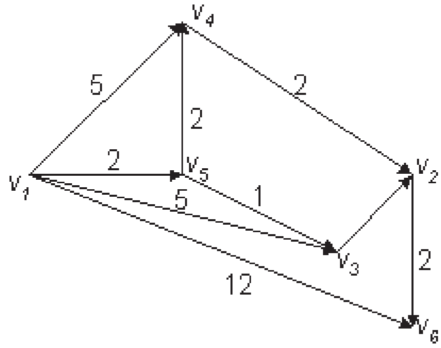


Рис. 1.21

Составим матрицу $C(D)$ длин дуг нагруженного орграфа D (табл. 1.3). Справа от матрицы $C(D)$ припишем шесть столбцов, которые будем определять, используя рекуррентное соотношение (1.2) и исходя из (1.1).

Величина $\lambda_6^{(5)} = 7$ выражает длину минимального пути из v_1 в v_6 в нагруженном орграфе D.

Найдем минимальное число $k_1 \geq 1$, при котором выполняется равенство

$\lambda_6^{(k_1)} = \lambda_6^{(5)}$. Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех минимальных путей из v_1 в v_6 в нагруженном графе D равняется 4.

Таблица 1.3

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	5	5	2	12		0	0	0	0	0	0
v_2	∞	∞	∞	∞	∞	2		∞	∞	7	5	5	5
v_3	∞	2	∞	∞	∞	∞		∞	5	3	3	3	3
v_4	∞	2	∞	∞	∞	∞		∞	5	4	4	4	4
v_5	∞	∞	1	2	∞	∞		∞	2	2	2	2	2
v_6	∞	∞	∞	∞	∞	∞		∞	12	12	9	7	7

Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5, где $i_1 = 6$, удовлетворяющих (1.4) (для этого используем формулу (1.2)).

Получаем, что в качестве такой последовательности надо взять номера 6, 2, 3, 5, 1, так как

$$\lambda_2^{(3)} + c_{2,6} = 5 + 2 = 7 = \lambda_6^{(4)};$$

$$\lambda_3^{(2)} + c_{3,2} = 3 + 2 = 5 = \lambda_2^{(3)};$$

$$\lambda_5^{(1)} + c_{5,3} = 2 + 1 = 3 = \lambda_3^{(2)};$$

$$\lambda_1^{(0)} + c_{1,5} = 0 + 2 = 2 = \lambda_5^{(1)}.$$

Тогда $v_1v_5v_3v_2v_6$ – искомый минимальный путь из v_1 в v_6 в нагруженном орграфе D, причем он содержит минимальное число дуг среди всех возможных минимальных путей из v_1 в v_6.

1.2.4. Эйлеровы цепи и циклы

Классической в теории графов является следующая задача. В городе Кенигсберге имеется два острова, соединенных семью мостами с берегами реки Преголь и друг с другом так, как показано на рис. 1.22. Задача состоит в следующем: осуществить прогулку по городу таким образом, чтобы, пройдя по одному разу по каждому мосту, вернуться обратно. Решение этой задачи сводится к нахождению некоторого специального маршрута в графе.

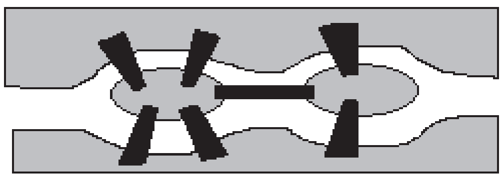


Рис. 1.22

Пусть G – псевдограф.

*Определение 1.52. Цепь (цикл) в G называется **эйлеровой (эйлеровым)**, если она (он) проходит по одному разу через каждое ребро псевдографа G.*

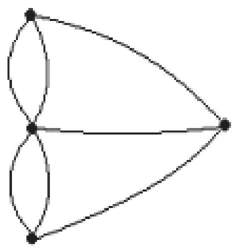


Рис. 1.23

Поставим в соответствие схеме мультиграф G, изображенный на рис. 1.23, в котором каждой части суши соответствует вершина, а каждому мосту – ребро, соединяющее соответствующие вершины. На языке теории графов задача звучит следующим образом: найти эйлеров цикл в мультиграфе G.

*Определение 1.53. Граф является **эйлеровым**, если он содержит эйлеров цикл.*

Рассмотрим вопрос о наличии эйлеровой цепи и цикла в псевдографе.

Теорема. Связный граф является эйлеровым тогда и только тогда, когда каждая вершина имеет четную локальную степень.

Теорема. Связный граф содержит эйлерову цепь тогда и только тогда, когда ровно две вершины имеют нечетную локальную степень.

Завершая рассмотрение эйлеровых графов, рассмотрим алгоритм построения эйлеровой цепи в данном эйлеровом графе. Этот метод известен под названием ***алгоритма Флёри***.

Теорема. Пусть G – эйлеров граф, тогда следующая процедура всегда возможна и приводит к эйлеровой цепи графа G. Выходя из произвольной вершины, идем по ребрам графа произвольным образом, соблюдая лишь следующие правила:

- *стираем ребра по мере их прохождения и стираем также изолированные вершины, которые при этом образуются;*
- *на каждом этапе идем по мосту только тогда, когда нет других возможностей.*

Любой простой полный граф с нечетным количеством вершин является эйлеровым. Любой циклический граф является эйлеровым. Граф, являющийся колесом, не является эйлеровым.

1.2.5. Гамильтоновы цепи и циклы

Пусть G – псевдограф.

*Определение 1.54. Цепь (цикл) в G называется **гамильтоновой (гамильтоновыми)**, если она (он) проходит по одному разу через каждую вершину псевдографа G.*

*Определение 1.55. Граф является **гамильтоновым**, если он содержит гамильтонов цикл.*

С понятием гамильтоновых циклов тесно связана так называемая задача коммивояжера: в нагруженном графе G определить гамильтонов цикл минимальной длины (иными словами, коммерсант должен совершить поездку по городам и вернуться обратно, побывав в каждом городе ровно один раз, и при этом стоимость такой поездки должна быть минимальной).

Приведем ***теорему Дирака***, которая отвечает на вопрос: существует ли в графе гамильтонов цикл.

Теорема. *Если в простом графе с* n (≥ 3) *вершинами локальная степень каждой вершины не менее* $n/2$, *то граф является гамильтоновым.*

Любой простой полный граф является гамильтоновым. Любой циклический граф является гамильтоновым. Граф, являющийся колесом, является гамильтоновым.

1.3. Деревья и циклы

1.3.1. Определение и свойства деревьев

Определение 1.56. *Граф G называется* ***деревом****, если он является связным и не имеет циклов. Граф G, все компоненты связности которого являются деревьями, называется* ***лесом****.*

Например, граф G_1, приведенный на рис. 1.24, *а*, является деревом, граф G_2, приведенный на рис. 1.24, *б*, – лесом, он содержит три связные компоненты, каждая из которых является деревом.

а) б)

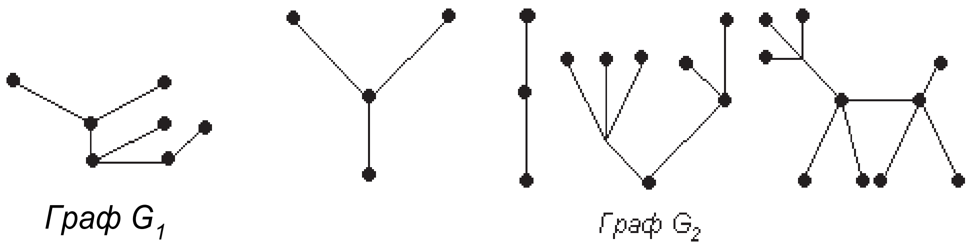


Рис. 1.24

Следующие утверждения эквивалентны:

- граф G есть дерево;
- граф G является связным и не имеет простых циклов;
- граф G является связным и число его ребер ровно на единицу меньше числа вершин;
- любые две различные вершины графа G можно соединить единственной (и притом простой) цепью;
- граф G не содержит циклов, но, добавляя к нему любое новое ребро, получаем ровно один (с точностью до направления обхода и начальной вершины обхода) и притом простой цикл (проходящий через добавляемое ребро).

Утверждения.

1) Если у дерева G есть, по крайней мере, одно ребро, то у него обязательно найдется висячая вершина.

2) Пусть G – дерево. Тогда любая цепь в G будет простой.

1.3.2. Остовное дерево связного графа

Определение 1.57. ***Остовным деревом*** *связного графа G называется любой его подграф, содержащий все вершины графа G и являющийся деревом.*

Пусть G – связный граф. Тогда остовное дерево графа G (если оно существует) должно содержать $n(G)$-1 ребер. Таким образом, любое остовное дерево графа G есть результат удаления из G ровно $m(G)-(n(G)-1) = m(G)-n(G)+1$ ребер.

Определение 1.58. Число $m(G)-n(G)+1$ называется ***цикломатическим числом*** *связного графа G и обозначается через $\nu(G)$.*

Замечание. Понятие остовного дерева и цикломатического числа аналогичным образом определяется и для произвольного связного псевдографа G.

Покажем существование остовного дерева для произвольного связного псевдографа $G = (V,X)$, описав алгоритм его выделения.

Шаг 1. Выбираем в G произвольную вершину u_1, которая образует подграф G_1 псевдографа G, являющийся деревом. Полагаем $i = 1$.

Шаг 2. Если $i = n$, где $n = n(G)$, то задача решена, и G_i – искомое остовное дерево псевдографа G. В противном случае переходим к шагу 3.

Шаг 3. Пусть уже построено дерево G_i, являющееся подграфом псевдографа G и содержащее некоторые вершины $u_1, \ldots, u_i$, где $1 \le i \le n-1$. Строим граф G_{i+1}, добавляя к графу G_i новую вершину $u_{i+1} \in V$, смежную в G с некоторой вершиной u_j графа G_i, и новое ребро (u_{i+1}, u_j) (в силу связности G и того обстоятельства, что $i < n$, указанная вершина u_{i+1} обязательно найдется). Согласно утверждению, граф G_{i+1} также является деревом. Присваиваем $i:=i+1$ и переходим к шагу 2.

Пример 1.8

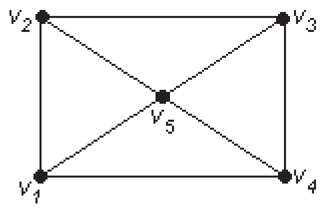


Рис. 1.25

Используя алгоритм, выделим остовное дерево графа G, изображенного на рис. 1.25.

Решение.

На рис. 1.26 приведена последовательность графов G_i, $i=1, 2, 3, 4, 5$, получаемых в результате выполнения алгоритма.

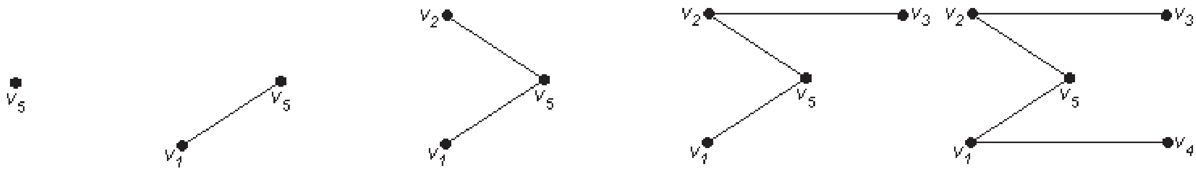


Рис. 1.26

При этом в силу того, что $n(G) = 5$, G_5 – остовное дерево графа G.

Замечание. Остовное дерево связного графа может быть выделено, вообще говоря, не единственным способом.

Общее число остовных деревьев связного графа может оказаться весьма большим. Например, для полного графа с n вершинами оно равно n^{n-2}.

В общем случае обозначим через G произвольный граф с n вершинами, m ребрами и k компонентами. Как уже было сказано, применяя вышеописанный алгоритм, получим в результате граф, являющийся остовным лесом.

Определение 1.59. Число ребер, удаленных при построении остовного леса произвольного графа G, называется ***цикломатическим числом*** *(или цикломатическим рангом) графа G и обозначается через* $\gamma(G) = m-n+k$.

Циклический ранг дерева равен нулю, а циклический ранг циклического графа равен единице.

Определение 1.60. ***Коцикличeским рангом*** *графа G называется число ребер в его остовном дереве.*

С понятием остовного леса T графа G тесно связано понятие фундаментальной системы циклов, ассоциированной с T.

Определение 1.61. Если добавить к T любое не содержащееся в нем ребро графа G, то получим единственный цикл; множество всех циклов, получаемых таким способом (т.е. путем добавления по отдельности каждого ребра из G, не содержащегося в T), называется ***фундаментальной системой циклов, ассоциированной с T****.*

1.3.3. Минимальные остовные деревья нагруженных графов

Пусть теперь каждому ребру $x \in X$ связного графа $G = (V,X)$ с непустым множеством ребер X поставлена в соответствие величина $l(x)$ – длина ребра x, т.е. граф G является нагруженным. Приведем алгоритм, позволяющий найти остовное дерево графа G с минимальной суммой длин содержащихся в нем ребер (по сравнению со всеми другими остовными деревьями графа G).

Определение 1.62. Остовное дерево связного нагруженного графа G с минимальной суммой длин содержащихся в нем ребер будем называть ***минимальным остовным деревом*** *(МОД) графа G.*

***Алгоритм выделения МОД нагруженного связного графа G*:**

Шаг 1. Выберем в графе G ребро минимальной длины. Вместе с инцидентными ему вершинами оно образует подграф G_2 графа G. Положим $i = 2$.

Шаг 2. Если $i = n$, где $n = n(G)$, то задача решена, и G_i – искомое МОД графа G. В противном случае переходим к шагу 3.

Шаг 3. Строим граф G_{i+1}, добавляя к графу G_i новое ребро минимальной длины, выбранное среди всех ребер графа G, каждое из которых инцидентно к какой-нибудь вершине графа G_i и одновременно инцидентно какой-нибудь вершине графа G, не содержащейся в G_i. Вместе с этим ребром включаем в G_{i+1} и инцидентные ему вершины, не содержащиеся в G_i. Присваиваем $i:=i+1$ и переходим к шагу 2.

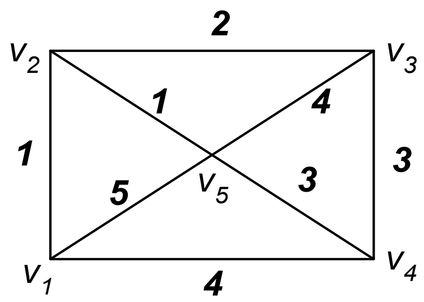


Рис. 1.27

Пример 1.9

Определить МОД нагруженного графа G, изображенного на рис. 1.27, используя алгоритм.

Решение.

На рис. 1.28 приведена последовательность графов G_i, $i = 2, 3, 4, 5$, получаемых в результате выполнения алгоритма. При этом в силу того, что $n(G) = 5$, G_5 – МОД графа G. Причем, МОД(G) = 7.

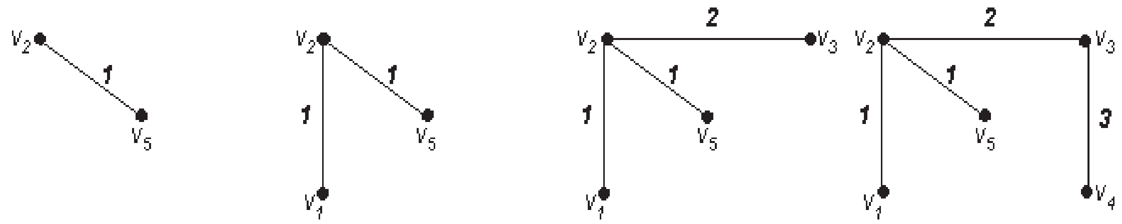


Рис. 1.28

Замечания.

1) Возможно выделить в графе несколько остовных деревьев, каждое из которых будет являться минимальным, при этом величины МОД для всех этих деревьев будут равными.

2) Для выделения МОД нагруженного псевдографа G следует предварительно удалить из G петли, из кратных ребер оставить лишь ребра минимальной длины.

1.4. Транспортные сети

Определение 1.63. ***Транспортной сетью*** *называется орграф* $D = (V, X)$ *с множеством вершин* V*, для которого выполняются условия:*

1) существует одна и только одна вершина v_1*, называемая* ***источником****, такая, что* $D^{-1}(v_1) = 0$ *(т.е. ни одна дуга не заходит в* v_1*);*

2) существует одна и только одна вершина v_n*, называемая* ***стоком****, такая, что* $D(v_n) = 0$ *(т.е. из* v_n *не исходит ни одной дуги);*

3) каждой дуге $x \in X$ *поставлено в соответствие целое число* $c(x) \geq 0$*, называемое* ***пропускной способностью дуги****.*

Определение 1.64. *Вершины в транспортной сети, отличные от источника и стока, называются* ***промежуточными****.*

1.4.1. Поток в транспортной сети

Определение 1.65. *Функция* $\varphi(x)$*, определенная на множестве* X *дуг транспортной сети* D *и принимающая целочисленные значения, называется* ***допустимым потоком*** *(или просто потоком) в транспортной сети* D*, если:*

1) для любой дуги $x \in X$ *величина* $\varphi(x)$*, называемая* ***потоком по дуге*** x*, удовлетворяет условию* $0 \leq \varphi(x) \leq c(x)$*;*

2) для любой промежуточной вершины v *выполняется равенство*

$$\sum_{\varpi \in D^{-1}(v)} \varphi(\omega, v) - \sum_{\varpi \in D(v)} \varphi(v, \omega) = 0,$$

т.е. сумма потоков по дугам, заходящим в v*, равна сумме потоков по дугам, исходящим из* v*.*

Определение 1.66. ***Величиной потока*** φ *в транспортной сети* D *называется величина* $\overline{\varphi}$*, равная сумме потоков по всем дугам, заходящим в* v_n*, или, что то же самое, величина, равная сумме потоков по всем дугам, исходящим из* v_1, т.е.

$$\overline{\varphi} = \sum_{v \in D^{-1}(v_n)} \varphi(v, v_n) = \sum_{v \in D(v_1)} \varphi(v_1, v)$$

Пусть φ – допустимый поток в транспортной сети D.

Определение 1.67. *Дуга* $x \in X$ *называется* ***насыщенной****, если поток по ней равен её пропускной способности, т.е. если* $\varphi(x) = c(x)$*. Поток* φ *называется* ***полным****, если любой путь в* D *из* v_1 *в* v_n *содержит, по крайней мере, одну насыщенную дугу.*

Определение 1.68. *Поток* φ *называется* ***максимальным****, если его величина принимает максимальное значение по сравнению с другими допустимыми потоками в транспортной сети* D*.*

Очевидно, что максимальный поток φ обязательно является полным. Обратное неверно. Существуют полные потоки, не являющиеся максимальными. Тем не менее, полный поток можно рассматривать как некоторое приближение к максимальному потоку.

Алгоритм построения полного потока в транспортной сети:

Шаг 1. Полагаем $\forall x \in X \varphi(x) = 0$ (т.е. начинаем с нулевого потока). Кроме того, полагаем $D' = D$.

Шаг 2. Удаляем из орграфа D' все дуги, являющиеся насыщенными при потоке φ в транспортной сети D. Полученный орграф снова обозначаем через D'.

Шаг 3. Ищем в D' простую цепь η из v_1 в v_n. Если такой цепи нет, то φ – искомый полный поток в транспортной сети D. В противном случае переходим к шагу 4.

Шаг 4. Увеличиваем поток $\varphi(x)$ по каждой дуге x из η на одинаковую величину $\boldsymbol{a} > 0$ такую, что, по крайней мере, одна дуга из η оказывается насыщенной, а потоки по остальным дугам из η не превышают их пропускных способностей. При этом величина потока также увеличивается на $\boldsymbol{a}$, а сам поток φ в транспортной сети D остается допустимым (поскольку в каждую промежуточную вершину, содержащуюся в η, дополнительно вошло $\boldsymbol{a}$ единиц потока и из нее вышло также $\boldsymbol{a}$ единиц потока). После этого переходим к шагу 2.

Пример 1.10

Построить полный поток в транспортной сети, изображенной на рис. 1.29.

Решение.

Начинаем с нулевого потока (рис. 1.30). Пошагово выделяем простые цепи и увеличиваем поток по ним таким образом, чтобы хотя бы одна дуга в каждой из них стала насыщенной. Полученную насыщенную дугу помечаем пунктиром, помня, что по насыщенной дуге больше идти нельзя.

1) $\eta_1 = v_1v_2v_4v_6$, $a_1 = \min\{7, 3, 7\} = 3$ (рис. 1.31).

2) $\eta_2 = v_1v_2v_3v_4v_6$, $a_2 = \min\{7\text{-}3, 2, 2, 7\text{-}3\} = 2$ (рис. 1.32).

3) $\eta_3 = v_1v_3v_5v_6$, $a_3 = \min\{8, 2, 9\} = 2$ (рис. 1.33).

4) $\eta_4 = v_1v_2v_5v_6$, $a_4 = \min\{7\text{-}5, 6, 9\text{-}2\} = 2$ (рис. 1.34).

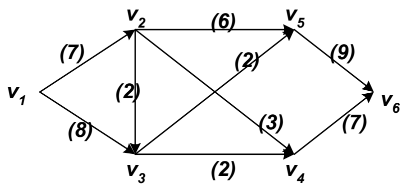


Рис. 1.29

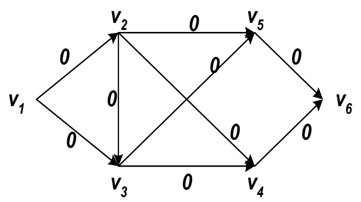


Рис. 1.30

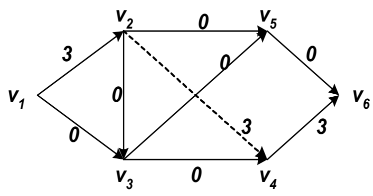


Рис. 1.31

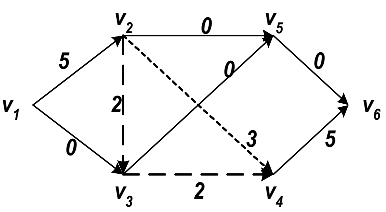


Рис. 1.32

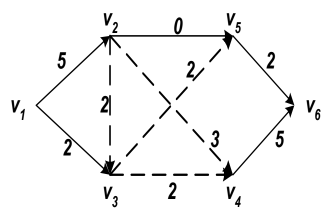


Рис. 1.33

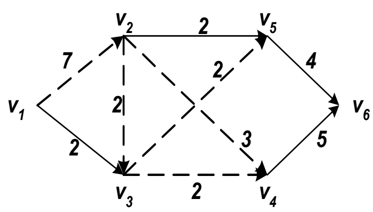


Рис. 1.34

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и φ = 3 + 2 + 2 + 2 = 9.

1.4.2. Орграф приращений

Выделим для заданной транспортной сети D и допустимого потока φ в этой сети орграф приращений $I(D,\ \varphi)$, имеющий те же вершины, что и сеть D. Каждой дуге $x = (v,\ \omega)\in X$ транспортной сети D в орграфе приращений $I(D,\ \varphi)$ соответствуют две дуги: x и $x' = (\omega, v)$ – дуга, противоположная по направлению дуге x. Припишем дугам $x = (v,\ \omega)\in X,\ x' = (\omega,\ v)$ орграфа приращений $I(D,\ \varphi)$ длину l:

$$l(x) = \begin{cases} 0,\ \text{если}\ \ \varphi(x) < c(x); \\ \infty,\ \text{если}\ \ \varphi(x) = c(x). \end{cases} \qquad l'(x) = \begin{cases} 0,\ \text{если}\ \ \varphi(x) > 0; \\ \infty,\ \text{если}\ \ \varphi(x) = 0, \end{cases}$$

т.е. орграф $I(D,\ \varphi)$ является нагруженным. При этом очевидно, что длина любого пути из v_1 в v_n в орграфе $I(D,\ \varphi)$ равна либо 0, либо ∞.

Пример 1.11

Построить орграф приращений для данной транспортной сети и построенного полного потока в этой сети (рис. 1.35).

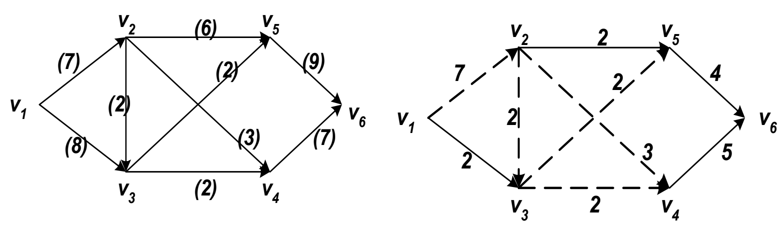


Рис. 1.35

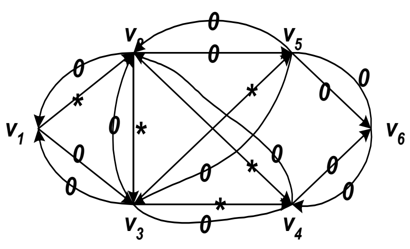


Рис. 1.36

Решение.

Напомним, что орграф приращений имеет в два раза больше дуг, чем исходная транспортная сеть. Для дуги прямой направленности вес равен 0, если дуга не является насыщенной, ∞ – в противном случае. Для дуги обратной направленности вес равен 0, если поток по ней не равен нулю, ∞ – в противном случае. На рис. 1.36 изображен орграф приращений, соответствующий данному потоку в исходной транспортной сети.

***Алгоритм построения максимального потока в транспортной сети*:**

Шаг 1. Полагаем $i = 0$. Пусть φ_0 – любой допустимый поток в транспортной сети D (например, полный; можно начинать с нулевого потока: $\varphi_0(x)$, $x \in X$).

Шаг 2. По сети D и потоку φ_i строим орграф приращений $I(D, \varphi_i)$.

Шаг 3. Находим простую цепь η_i, являющуюся минимальным путем из v_1 в v_n в нагруженном орграфе $I(D, \varphi_i)$ (например, используя алгоритм Форда-Беллмана). Если длина этой цепи равна ∞, то поток φ_i максимален, и работа алгоритма закончена. В противном случае увеличиваем поток вдоль цепи η_i на максимально допустимую величину $a_i > 0$, где $a_i \in Z$ (прибавляя ее для каждой дуги $x \in X$, через которую проходит цепь η_i, к уже имеющейся величине потока по дуге x, если направления x и η_i совпадают, и, вычитая, если направления x и η_i противоположны).

Пример 1.12

Выяснить является ли полный поток максимальным (рис. 1.37), если нет, то дополнить его до максимального.

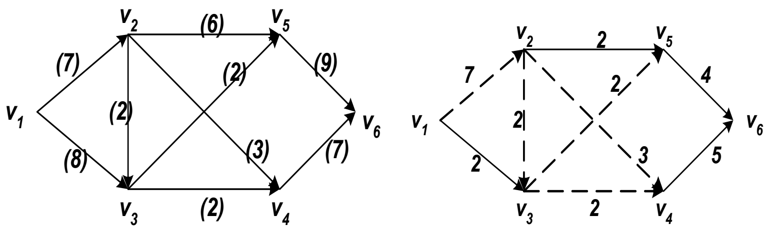


Рис. 1.37

Решение.

Для решения используем алгоритм Форда-Беллмана нахождения минимального пути в нагруженном орграфе.

Построим матрицу длин дуг $C(D)$ и λ-матрицу (табл. 1.4).

Таблица 1.4

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	0	∞	∞	∞		0	0	0	0	0	0
v_2	0	∞	∞	∞	0	∞		∞	∞	0	0	0	0
v_3	0	0	∞	0	0	∞		∞	0	0	0	0	0
v_4	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	0
v_5	∞	0	0	∞	∞	0		∞	∞	∞	0	0	0
v_6	∞	∞	∞	0	0	∞		∞	∞	∞	∞	0	0

Поскольку $\lambda_6^{(5)} = 0$, то существует нулевой путь из источника v_1 в сток v_6. Значит, полный поток не является максимальным. Дополним его до максимального. Для этого найдем путь нулевой длины: $\lambda_6^{(5)} = 0 = \lambda_6^{(4)}$.

Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех нулевых путей из v_1 в v_6 в орграфе приращений равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5, где $i_1 = 6$.

Получаем, что в качестве такой последовательности надо взять номера 1, 3, 2, 5, 6, так как

$$\lambda_5^{(3)} + c_{5,6} = 0 + 0 = 0 = \lambda_6^{(4)}; \qquad \lambda_3^{(1)} + c_{3,2} = 0 + 0 = 0 = \lambda_2^{(2)};$$

$$\lambda_2^{(2)} + c_{2,5} = 0 + 0 = 0 = \lambda_5^{(3)}; \qquad \lambda_3^{(0)} + c_{1,3} = 0 + 0 = 0 = \lambda_3^{(1)}.$$

Тогда $v_1v_3v_2v_5v_6$ – искомый нулевой путь из v_1 в v_6. Дуги, совпадающие по направлению с дугами исходной транспортной сети, помечаем знаком «+», не совпадающие – знаком «-».

Получаем, $v_1 \underset{+}{\rightarrow} v_3 \underset{-}{\rightarrow} v_2 \underset{+}{\rightarrow} v_5 \underset{+}{\rightarrow} v_6$. Теперь необходимо найти величину, которую будем перемещать по полученному контуру. Для этого каждому ребру в контуре поставим в соответствие число $\alpha(i, j)$, которое находим по следующему правилу: если направление ребра (i, j) в контуре совпадает с направлением ребра x в транспортной сети, то $\alpha(i, j) = c(x)$-$\varphi(x)$; если направление ребра в контуре не совпадает с направлением ребра в транспортной сети, то $\alpha(i, j) = \varphi(x)$. Итак, из чисел $\alpha(i, j)$ найдем минимальное: $\min\{8 - 2 = 6, 2, 6 - 2 = 4, 9 - 4 = 5\} = 2$.

Перемещаем по контуру 2. В результате получаем поток, изображенный на рис. 1.38.

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и $\varphi = 11$. Проверим, является ли он максимальным.

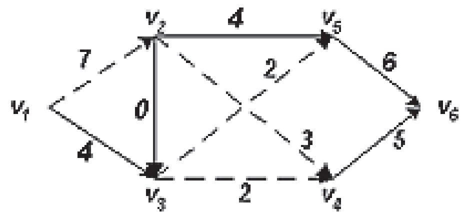


Рис. 1.38

Построим матрицу длин дуг $C(D)$ и λ-матрицу (табл. 1.5).

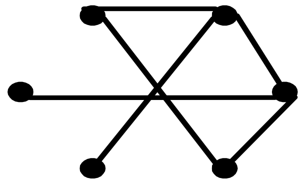
Таблица 1.5

	v_1	v_2	v_3	v_4	v_5	v_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
v_1	∞	∞	0	∞	∞	∞		0	0	0	0	0	0
v_2	0	∞	0	∞	0	∞		∞	∞	∞	∞	∞	∞
v_3	0	∞	∞	∞	∞	∞		∞	0	0	0	0	0
v_4	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	∞
v_5	∞	0	0	∞	∞	0		∞	∞	∞	∞	∞	∞
v_6	∞	∞	∞	0	0	∞		∞	∞	∞	∞	∞	∞

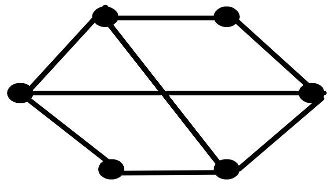
Так как $\lambda_6^{(5)} = \infty$, то нулевого пути из v_1 в v_6 не существует. Значит, поток $\overline{\varphi} = 11$ является максимальным.

Проверочный тест по теме «Теория графов»

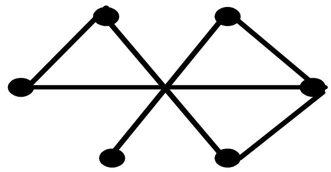
Вопрос 1. Чему равно цикломатическое число графа?



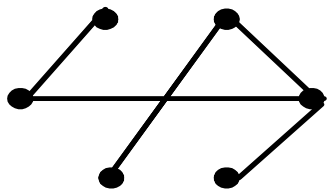
Вопрос 2. Чему равно цикломатическое число графа?



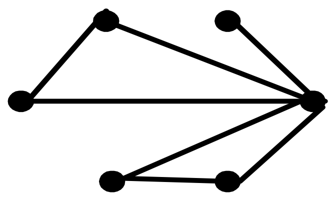
Вопрос 3. Чему равно цикломатическое число графа)



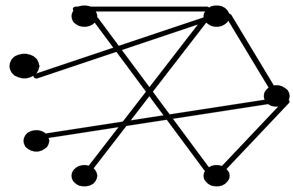
Вопрос 4. Чему равно цикломатическое число графа?



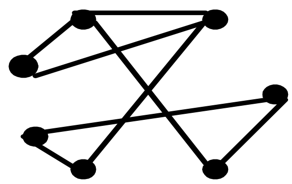
Вопрос 5. Чему равно цикломатическое число графа?



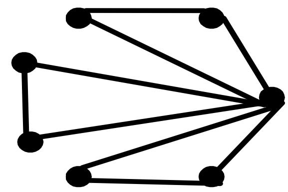
Вопрос 6. Чему равно цикломатическое число графа?



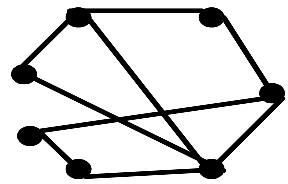
Вопрос 7. Чему равно цикломатическое число графа?



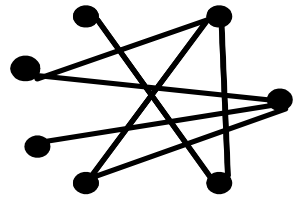
Вопрос 8. Чему равно цикломатическое число графа?



Вопрос 9. Чему равно цикломатическое число графа?



Вопрос 10. Чему равно цикломатическое число графа?



Вопрос 11. Укажите графы, которые всегда являются эйлеровыми:
а) любой простой полный граф;
б) любой простой полный граф с нечетным количеством вершин;
в) любой циклический граф;
г) колесо.

Вопрос 12. Какая из нижеперечисленных формул для цикломатического числа является правильной?
а) $\upsilon = m - n - k$;
б) $\upsilon = m - n + k$;
в) $\upsilon = m + n + k$;
г) $\upsilon = m + n - k$.

Вопрос 13. Чему равен $FW_3(V_1)$?

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
V_1	0	1	1	1	0	0	0	0
V_2	1	0	0	1	0	0	1	0
V_3	1	0	0	1	0	1	0	0
V_4	1	1	1	0	1	1	0	0
V_5	0	0	0	1	0	1	0	1
V_6	0	0	1	1	1	0	0	0
V_7	0	1	0	0	0	0	0	1
V_8	0	0	0	0	1	0	1	0

а) $FW_3(V_1) = \{V_4, V_5, V_8\}$;
б) $FW_3(V_1) = \{V_6, V_8\}$;
в) $FW_3(V_1) = \{V_6\}$;
г) $FW_3(V_1) = \{V_8\}$.

Вопрос 14. Какое из утверждений может быть взято в качестве определения дерева?

а) связный, ацикличный граф;

б) граф, не имеющий циклов;

в) любой граф, число ребер которого на единицу меньше числа вершин;

г) граф, любые две различные вершины которого можно соединить единственной простой цепью.

Вопрос 15. Как называется вершина в транспортной сети, из которой не исходит ни одной дуги?

а) сток;

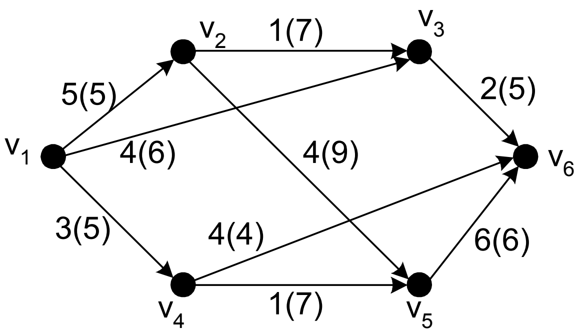
б) источник;

в) изолированная вершина;

г) промежуточная вершина.

Вопрос 16. Возможно ли выделить в нагруженном графе несколько остовных деревьев, каждое из которых будет являться минимальным?

Вопрос 17. Перечислить насыщенные дуги в транспортной сети, изображённой на рисунке.



а) V_1V_2;

б) V_4V_6;

в) V_1V_2, V_4V_6, V_5V_6;

г) V_1V_3.

Вопрос 18. Чему равен $FW_1(V_1)$, исходя из данных таблицы?

	V_1	V_2	V_3	V_4	V_5	V_6	V_7
V_1	0	1	0	1	0	0	0
V_2	1	0	1	0	1	0	0
V_3	0	1	0	0	1	0	1
V_4	1	0	0	0	1	1	0
V_5	0	1	1	1	0	1	0
V_6	0	0	0	1	1	0	1
V_7	0	0	1	0	0	1	0

а) $FW_1(V_1) = \{V_1, V_3, V_5, V_6, V_7\}$;

б) $FW_1(V_1) = \{V_2\}$;

в) $FW_1(V_1) = \{V_2, V_4\}$;

г) $FW_1(V_1) = \{V_4\}$.

Вопрос 19. Как называется вершина V_n транспортной сети, если $D^{-1}(V_n) = \varnothing$?

а) сток; в) висячая вершина;

б) источник; г) промежуточная вершина.

Вопрос 20. Укажите верные высказывания:

а) максимальный поток обязательно является полным;

б) полный поток обязательно является максимальным;

в) если путь в D из V_1 в V_n содержит хотя бы одну насыщенную дугу, то поток максимальный;

г) если путь в D из V_1 в V_n содержит хотя бы одну насыщенную дугу, то поток полный.

Вопрос 21. Вершина и ребро инцидентны, если:

а) они ориентированны;

б) они являются параллельными;

в) вершина является концом или началом ребра.

Вопрос 22. Для чего применяется алгоритм «Фронт волны»?

а) для поиска минимального пути из вершины V_1 в вершину V_n;

б) для поиска кратчайшего пути из вершины V_1 в вершину V_n;

в) для поиска максимального пути из вершины V_1 в вершину V_n;

г) для поиска пути из вершины V_1 в вершину V_n.

Вопрос 23. Если $\lambda_{i_1}^{(n-1)} < \infty$, тогда число $\lambda_{i_1}^{(n-1)}$ обозначает:

а) выражение суммы вершин и ребер;

б) выражает число минимальных путей;

в) выражает величину максимального пути из v_1 в v_{i_1} в нагруженном орграфе;

г) выражает величину минимального пути из v_1 в v_{i_1} в нагруженном орграфе;

д) выражает вес любого пути.

Вопрос 24. Чему равна величина потока в транспортной сети?

а) величине, равной сумме потоков по всем дугам, заходящим в сток;

б) величине, равной сумме потоков по дугам, исходящим из источника, плюс сумма потоков, заходящих в сток;

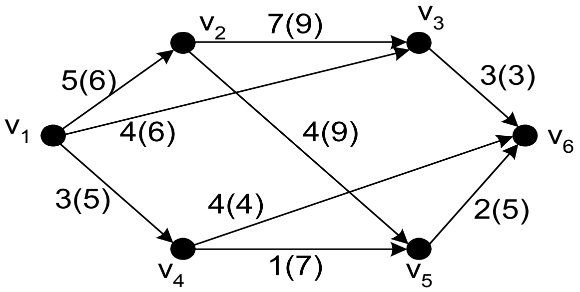
в) величине, равной сумме потоков по всем дугам, исходящим из источника;

г) величине, равной сумме потоков по дугам, исходящим из источника, минус сумма потоков, заходящих в сток.

Вопрос 25. Циклический ранг дерева равен:
а) 0;
б) 1;
в) 2.

Вопрос 26. Граф называется ориентированным, если:
а) одно его ребро ориентированно;
б) каждое ребро ориентированно;
в) нет ориентированных рёбер.

Вопрос 27. Определить максимальную пропускную способность ребра в транспортной сети, изображенной на рисунке.



а) 5;
б) 7;
в) 9;
г) 3.

Вопрос 28. По таблице определить, чему равен радиус графа?

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$D(V_i)$
V_1	0	1	1	1	2	2	2	3	3
V_2	1	0	2	1	2	2	1	2	2
V_3	1	2	0	1	2	1	3	3	3
V_4	1	1	1	0	1	1	2	2	2
V_5	2	2	2	1	0	1	2	1	2
V_6	2	2	1	1	1	0	3	2	3
V_7	2	1	3	2	2	3	0	1	3
V_8	3	2	3	2	1	2	1	0	3

а) 0; в) 2;
б) 1; г) 3.

Вопрос 29. Образ вершины $D(V_i)$ – это:

а) множество вершин, из которых можно попасть в заданную за один шаг;

б) множество вершин, в которые можно попасть из данной за один шаг;

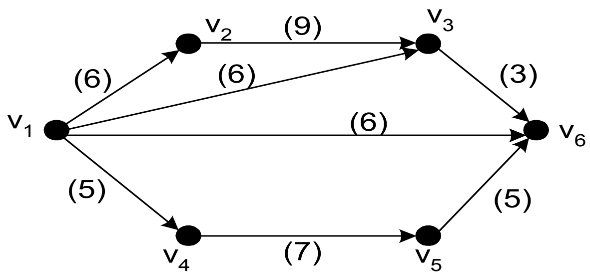
г) множество вершин, из которых можно попасть в данную за несколько шагов.

Вопрос 30. Все ли цепи в дереве являются простыми?

Вопрос 31. Как называется граф G, все компоненты связности которого являются деревьями?

Вопрос 32. Циклический граф – это:
а) неоднородный степени 4 с *n* вершинами;
б) граф, у которого любые две вершины смежные;
в) однородный степени 2 с *n* вершинами;
г) граф, у которого любая вершина является изолированной.

Вопрос 33.Сколько путей из источника в сток существует в данной сети?



Вопрос 34. Для каких графов применяется алгоритм «Фронт-волны»?
а) для ориентированных графов;
б) для неориентированных графов;
д) для ориентированных и неориентированных графов.

Вопрос 35. По таблице определить, чему равен диаметр графа?

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$D(V_i)$
V_1	0	1	1	1	2	2	2	3	3
V_2	1	0	2	1	2	2	1	2	2
V_3	1	2	0	1	2	1	3	3	3
V_4	1	1	1	0	1	1	2	2	2
V_5	2	2	2	1	0	1	2	1	2
V_6	2	2	1	1	1	0	3	2	3
V_7	2	1	3	2	2	3	0	1	3
V_8	3	2	3	2	1	2	1	0	3

а) 0;
б) 1;
в) 2;
г) 3.

Вопрос 36. Что является дополнением простого полного графа K_n?
а) простой полный граф с *n* вершинами;
б) полный двудольный граф с *n* вершинами;
в) вполне несвязный граф с *n* вершинами.

Вопрос 37. Циклом называется:

а) граф, который содержит больше одного разреза;

б) замкнутая простая цепь;

в) множество рёбер, удаление которых приводит к увеличению числа связных компонентов;

г) маршрут, если все его рёбра различны.

Вопрос 38. Какой путь называется минимальным?

а) если произведение весов ребер, составляющих этот путь, является наименьшим по сравнению с другими путями из первой в последнюю вершину;

б) если сумма весов ребер, составляющих этот путь, является наименьшей по сравнению с другими путями;

в) если сумма вершин является пустым множеством;

г) если произведение вершин является наименьшим;

д) если полусумма весов ребер составляет наименьший путь по сравнению с другими путями.

Вопрос 39. По таблице выяснить, какие из вершин являются центрами графа?

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$D(V_i)$
V_1	0	1	1	1	2	2	2	3	3
V_2	1	0	2	1	2	2	1	2	2
V_3	1	2	0	1	2	1	3	3	3
V_4	1	1	1	0	1	1	2	2	2
V_5	2	2	2	1	0	1	2	1	2
V_6	2	2	1	1	1	0	3	2	3
V_7	2	1	3	2	2	3	0	1	3
V_8	3	2	3	2	1	2	1	0	3

а) V_2, V_4, V_5; в) V_1, V_8;

б) V_1, V_3, V_6, V_7, V_8; г) V_1.

Вопрос 40. Какой будет любая цепь в графе *G*, если *G* – дерево?

Вопрос 41. Диаметром связного графа называется:

а) такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных;

б) количество его рёбер;

в) максимально возможное расстояние между двумя его вершинами.

Вопрос 42. Цикломатическое число графа показывает:

а) сколько ребер необходимо удалить, чтобы граф стал деревом;

б) сколько ребер необходимо добавить, чтобы граф стал деревом;

в) сколько вершин и инцидентных им ребер необходимо добавить, чтобы граф стал деревом;

г) сколько вершин и инцидентных им ребер необходимо удалить, чтобы граф стал деревом.

Вопрос 43. Остовным деревом связного графа G называется:

а) любой его подграф, содержащий все вершины графа G;

б) любой его подграф, содержащий все вершины графа G и являющийся деревом;

в) любой его подграф, содержащий все вершины графа G и не являющийся деревом;

д) любой его подграф, являющийся деревом.

Вопрос 44. Дуга называется насыщенной, если:

а) поток по ней равен самому большему числу в сравнении с остальными;

б) поток по ней равен ее пропускной способности;

в) поток по ней равен максимальному потоку в транспортной сети.

Вопрос 45. Укажите графы, которые всегда являются гамильтоновыми:

а) любой простой полный граф;

б) любой простой полный граф с нечетным количеством вершин;

в) любой циклический граф;

г) колесо.

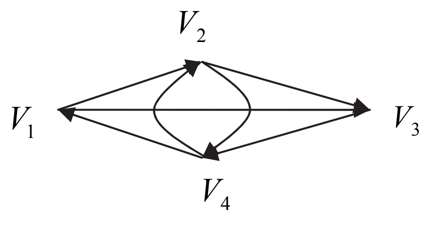
Вопрос 46. По рисунку определить, сколько в графе существует путей, длина которых равна 1?

а) 9;

б) 4;

в) 5;

г) 7.



Вопрос 47. Если на первом этапе нахождения кратчайшего пути из вершины V_1 в вершину V_n $FW_i(V_1) = \varnothing$, то это означает, что:

а) искомый путь содержит одно ребро;

б) искомого пути не существует;

в) искомый путь содержит все ребра данного графа;

г) искомый путь существует, но определить число его ребер невозможно.

Вопрос 48. Если минимального пути нет, то по каким параметрам это можно определить?

а) $k = 0$; в) $i = 0$; д) $\lambda_i^{(k)} = \infty$

б) $\lambda_n^{n-1} = 0$ г) $\lambda_n^{(n-1)} = \infty$

Вопрос 49. Что называется кратчайшим путем из вершины V_i в вершину V_j?

а) путь, содержащий наименьшее из возможных количество ребер;
б) путь, содержащий половину ребер данного графа;
в) путь, содержащий наибольшее из возможных количество ребер;
г) путь, содержащий все вершины данного графа.

Вопрос 50. Какой граф называется нагруженным?

а) если каждому ребру соответствует дробное положительное число;
б) если каждой вершине соответствует целое положительное число;
в) если для любых двух вершин существует простая цепь их связывающая;
г) если множество элементов (вершин и ребер) конечно и пусто, если его множество вершин пусто;
д) если каждому ребру соответствует целое положительное число.

Вопрос 51. Что обозначает величина λ_i^k в алгоритме Форда–Беллмана?

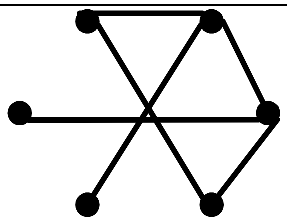
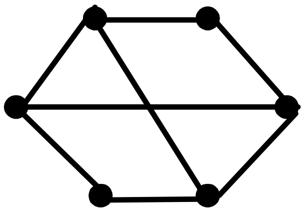
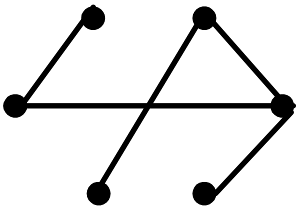
а) это максимальная величина пути из первой вершины в последнюю и содержит k вершин и i ребер;
б) это минимальная величина пути из первой в i-ю вершину и содержит не более k ребер;
в) это минимальная величина пути из первой в последнюю вершину и содержит i вершин и k ребер;
г) это минимальная величина пути из первой в предпоследнюю и содержит k вершин и i ребер;
д) это максимальная величина пути из первой в последнюю вершину и содержит i вершин и k ребер.

Вопрос 52. Какой поток называется полным?

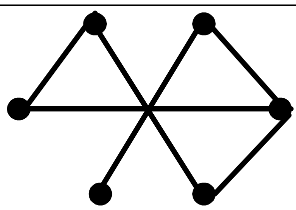
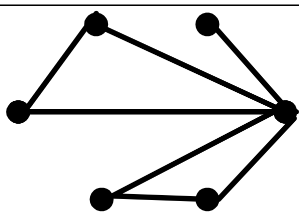
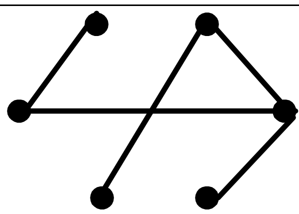
а) если любой путь из источника в сток содержит, по крайней мере, одну насыщенную дугу;
б) если путь из источника в сток проходит по всем ребрам в транспортной сети;
в) если в любом пути из источника в сток все дуги насыщенные;
г) если он содержит хотя бы один путь из источника в сток.

Вопрос 53. Какое остовное дерево называется минимальным?
а) дерево, содержащее минимальное количество ребер;
б) дерево с минимальной суммой весов содержащихся в нем ребер;
в) дерево с максимальной суммой весов содержащихся в нем ребер;
г) дерево, содержащее минимальное количество вершин.

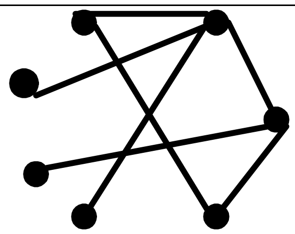
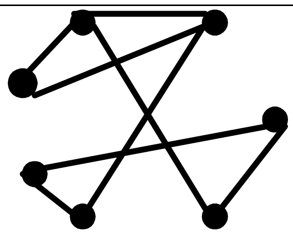
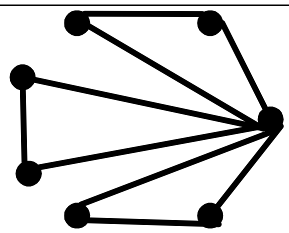
Вопрос 54. Установить соответствие между представлением графа и его цикломатическим числом:

 1	а) 1
 2	б) 2
 3	в) 0

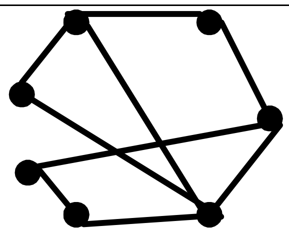
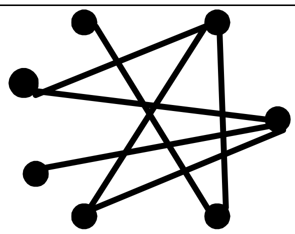
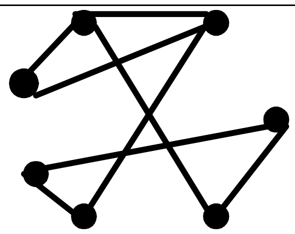
Вопрос 55.Установить соответствие между представлением графа и его цикломатическим числом:

 1	а) 1
 2	б) 2
 3	в) 0

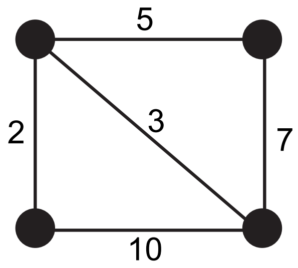
Вопрос 56. Установить соответствие между представлением графа и его цикломатическим числом:

 1	а) 1
 2	б) 2
 3	в) 3

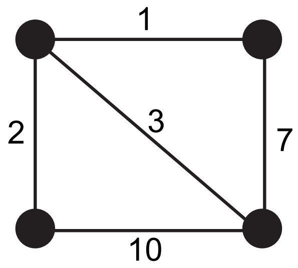
Вопрос 57. Установить соответствие между представлением графа и его цикломатическим числом:

 1	а) 3
 2	б) 1
 3	в) 2

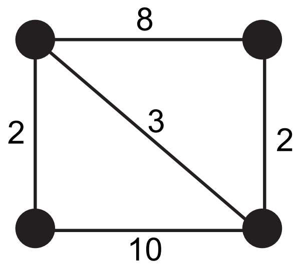
Вопрос 58. Чему равно минимальное остовное дерево графа?



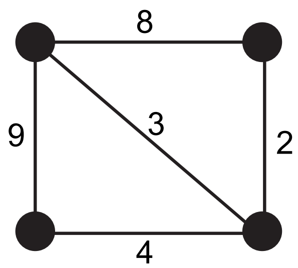
Вопрос 59. Чему равно минимальное остовное дерево графа?



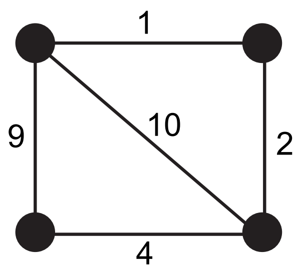
Вопрос 60. Чему равно минимальное остовное дерево графа?



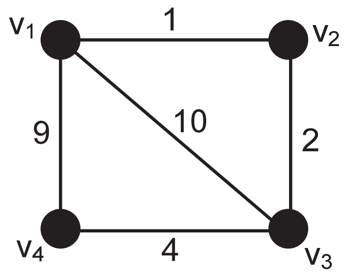
Вопрос 61. Чему равно минимальное остовное дерево графа?



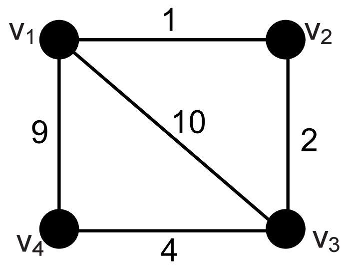
Вопрос 62. Чему равно минимальное остовное дерево графа?



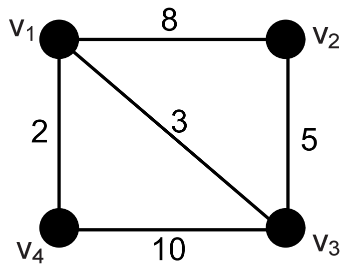
Вопрос 63. В каком порядке необходимо добавлять ребра для построения минимального остовного дерева, изображенного на рисунке?



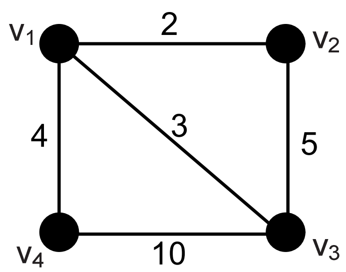
Вопрос 64. В каком порядке необходимо добавлять ребра для построения минимального остовного дерева, изображенного на рисунке?



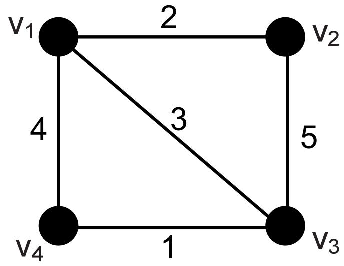
Вопрос 65. В каком порядке необходимо добавлять ребра для построения минимального остовного дерева, изображенного на рисунке?



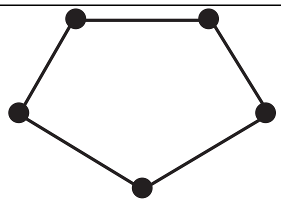
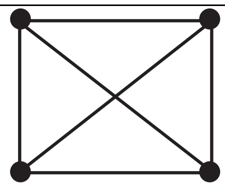
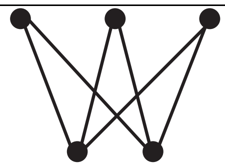
Вопрос 66. В каком порядке необходимо добавлять ребра для построения минимального остовного дерева, изображенного на рисунке?



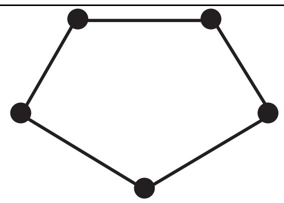
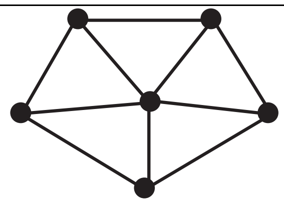
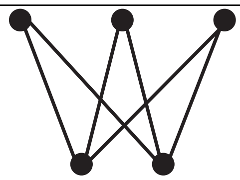
Вопрос 67. В каком порядке необходимо добавлять ребра для построения минимального остовного дерева, изображенного на рисунке?



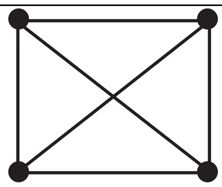
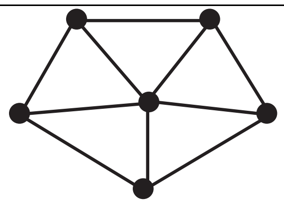
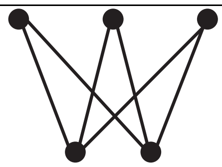
Вопрос 68. Установить соответствие между представлением графа и его названием:

 1.	а) циклический
 2.	б) простой полный
 3.	в) полный двудольный

Вопрос 69. Установить соответствие между представлением графа и его названием:

 1.	а) циклический
 2.	б) колесо
 3.	в) полный двудольный

Вопрос 70. Установить соответствие между представлением графа и его названием:

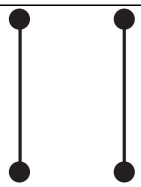
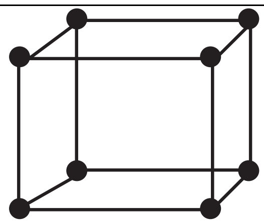
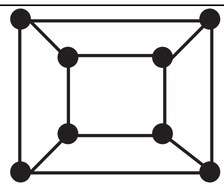
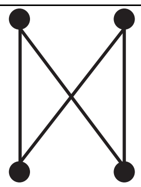
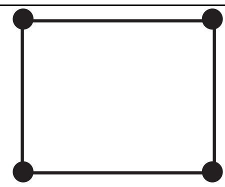
1. 	а) простой полный
2. 	б) колесо
3. 	в) полный двудольный

Вопрос 71. Дуга в транспортной сети называется ………., если поток по ней равен ее пропускной способности.

Вопрос 72. Связный граф содержит эйлерову цепь тогда и только тогда, когда ровно ……. вершины имеют нечетную локальную степень.

Вопрос 73. Связный граф является эйлеровым тогда и только тогда, когда каждая вершина имеет ………… локальную степень.

Вопрос 74. Установите соответствие между изоморфными графами:

1. 	а) 
2. 	б) 
3. 	в) 

Вопрос 75. В каком порядке выполняются шаги алгоритма «Фронт волны»:

а) строим матрицу смежности графа;

б) выясняем, существует ли путь из первой вершины в последнюю;

в) находим порядок вершин, составляющих искомый путь.

Вопрос 76. Используя алгоритм Форда–Беллмана, укажите порядок вершин, составляющих минимальный путь из V_1 в V_6:

	V_1	V_2	V_3	V_4	V_5	V_6		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$
V_1	∞	∞	5	5	2	12		0	0	0	0	0	0
V_2	∞	∞	∞	∞	∞	2		∞	∞	7	5	5	5
V_3	∞	2	∞	∞	∞	∞		∞	5	3	3	3	3
V_4	∞	2	∞	∞	∞	∞		∞	5	4	4	4	4
V_5	∞	∞	1	2	∞	∞		∞	2	2	2	2	2
V_6	∞	∞	∞	∞	∞	∞		∞	12	12	9	7	7

а) V_1; г) V_2;

б) V_5; д) V_6.

в) V_3;

1.5. Сетевое планирование и управление

Предположим, что строится современная школа. При этом выполняется сложный комплекс работ. Разные участки работ этого комплекса поручаются отдельным специалистам, организациям, бригадам, цехам, мастерским и т.д. Как наилучшим образом организовать отдельные работы, чтобы строительство закончить в наиболее короткий срок? Как распределить рабочую силу, материалы, финансы, оборудование, чтобы вся работа обошлась максимально дешево? Как поступить, если в процессе выполнения работ окажется, что какие-то исполнители не укладываются в срок? Как быстрее узнать, где в данный момент самый ответственный участок? С какого участка в данный момент можно взять людей, не срывая сроков выполнения всей работы? При планировании такого комплекса работ и руководстве им одной интуиции руководителя недостаточно.

В этом разделе будет показано, как построить сеть сложного комплекса работ, как определить по сети самые ответственные работы, как вычислить время завершения всего комплекса, как найти резервы времени для отдельных событий и работ.

1.5.1. Сетевой график

Проект (или комплекс работ) расчленяется на отдельные работы. Назовем хотя бы некоторые из работ, которые входят в комплекс работ по строительству школы. Они перечисляются здесь не в той последовательности, в которой должны выполняться.

- подвоз материалов, необходимых для строительства (песка, бетона, кирпичей и т. п.);
- геодезическая съемка местности;
- расчистка площадки для строительства;
- разработка проекта здания школы;
- монтаж фундамента;
- рытье котлована для фундамента;
- кладка стен;
- внутренняя электропроводка;
- доставка на стройплощадку блоков подъемного крана;
- принятие решения о строительстве школы;
- штукатурка стен и др.

Каждая отдельная работа, входящая в комплекс (проект), требует затрат определенного времени. Некоторые работы могут выполняться только в определенном порядке. Нельзя, например, укладывать фундамент, если еще не вырыт котлован для него. Существуют работы, входящие в комплекс, которые могут выполняться независимо друг от друга, одновременно. Например, при строительстве здания одновременно можно укладывать фундамент, завозить металлоконструкции для каркаса здания, монтировать подъемный кран.

При выполнении комплекса работ всегда можно выделить ряд событий, то есть итогов какой-то деятельности, позволяющих приступить к выполнению следующих работ. Например, проект утвержден; площадка для строительства расчищена; котлован вырыт; фундамент установлен.

Определение 1.69. *Под* ***событием*** *в сетевом планировании понимают завершение какой-либо работы (деятельности).*

Различают следующие разновидности событий сетевого графика:

1) ***Исходное*** *событие – начало выполнения проекта. Исходное событие не имеет предшествующих работ.*

2) ***Завершающее*** *событие – достижение конечной цели проекта (или одной из конечных целей). Завершающее событие не имеет следующих за ним работ.*

3) ***Промежуточное*** *событие (итог какой-то деятельности) – результат выполнения одной или нескольких работ, позволяющий приступить к выполнению последующих работ.*

*4) **Начальное** событие – событие, непосредственно предшествующее данной конкретной работе.*

*5) **Конечное** событие – событие, непосредственно следующее за данной работой.*

Событие не является процессом, оно не сопровождается затратами рабочей силы, времени и средств. Событие не может наступить, пока не закончатся все предшествующие ему работы.

Определение 1.70. *Под **работой** в сетевом планировании понимается:*

*1) **Действительная** работа – любой трудовой процесс, требующий затрат труда, времени и материальных ресурсов. (Примеры: проектирование двигателя, испытание его, монтаж фундамента, побелка потолков).*

*2) **Ожидание** – пассивный процесс, не требующий затрат труда и материальных ресурсов, но требующий затрат времени. (Примеры: твердение бетона, сушка штукатурки).*

*3) **Фиктивная** работа – чисто условная зависимость между событиями, которая вводится только для удобства изображения сети, фиктивная работа не связана с затратой труда, времени и ресурсов.*

Определение 1.71. *Ориентированный граф, в котором каждой вершине соответствует событие, а каждому ребру соответствует работа, называется **сетью**. Сеть называется **сетевым графиком**, если выполняются условия:*

1) существует единственная вершина, соответствующая исходному событию;

2) существует единственная вершина, соответствующая завершающему событию;

3) каждому ребру соответствует целое неотрицательное число, равное времени, которое необходимо для завершения соответствующей работы.

Сетевой график является графической моделью всего комплекса работ или производственного процесса. Он отражает взаимосвязь всех работ, событий, технологического процесса, обеспечение комплекса материальными и техническими ресурсами.

На сетевом графике ***событие*** изображается кружком (вершина графа), в котором проставляется число – шифр данного события. Фрагмент сетевого графика изображен на рис. 1.39. Все события на рис. 1.39 обозначены разными числами. Расшифровать их можно так:

0 – исходное событие, начало строительства;

1 – котлован подготовлен;

2 – монтаж фундамента закончен;

3 – металлоконструкции завезены;

4 – подъемный кран смонтирован.

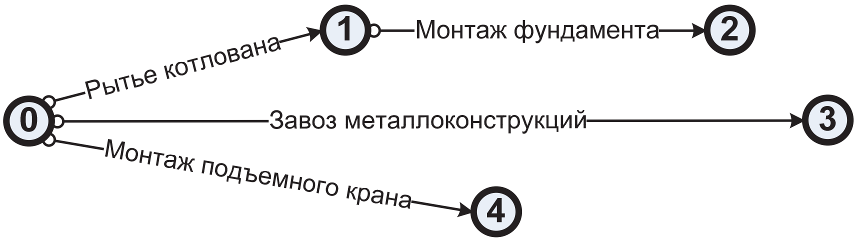


Рис. 1.39

Некоторые работы на рис. 1.39 выполняются в определенной последовательности, другие – параллельно.

Иногда для начала какой-то работы требуется завершение нескольких работ. Например, для начала монтажа требуется, чтобы был заложен фундамент, завезены металлоконструкции для каркаса здания и смонтирован подъемный кран. Эти работы выполняют разные люди, и завершиться они могут в разное время.

На сетевом графике действительная работа и ожидание изображаются сплошными стрелками, а фиктивная работа – штриховыми стрелками. Штриховые стрелки (рис. 1.40) отражают условную зависимость между событиями.

Рис. 1.40

Любая стрелка на сетевом графике соединяет только две вершины и отражает процесс перехода от одного события к другому. Поэтому любая работа может быть зашифрована парой чисел, соответствующих предшествующему и последующему событиям.

1.5.2. Построение сетевого графика

Прежде, чем непосредственно приступать к построению сетевого графика, составляют список всех работ, необходимых для выполнения заданного комплекса работ. Далее выясняют технологическую последовательность их выполнения; строят отдельные фрагменты сетевого графика; по возможности их упрощают. После этого из отдельных фрагментов строят общий сетевой график. В таких случаях принято говорить о *«сшивании сетевого графика»*.

Познакомимся с основными ***правилами построения*** *сетевых графиков*.

Правило 1. Каждую стрелку в сетевом графике по возможности рисуют так, чтобы ее конец находился правее начала, по возможности горизонтально.

Правило 2. Для удобства сетевой график строят без лишних пересечений стрелок. (Вместе с тем при составлении чернового варианта не следует увлекаться внешним видом сети)

Правило 3. Следят за тем, чтобы во все вершины, кроме той, которая соответствует исходному событию, входила, по меньшей мере, одна стрелка, так как все события, кроме исходного, имеют предшествующую работу.

Правило 4. Следят за тем, чтобы из всех вершин сети, кроме той, которая соответствует завершающему событию, выходили стрелки, так как все события, кроме завершающего, имеют последующую работу.

Правило 5. Следят за тем, чтобы в сетевом графике не образовывалось циклов.

Правило 6. Если одно событие служит началом для двух или более работ, после завершения которых начинается выполнение следующей работы, то вводится штриховая стрелка (условная зависимость) и дополнительное событие со своим номером.

На рис. 1.41 приведены неправильное и правильное изображения двух работ *a* и *b,* начинающихся после события 1; после завершения *a* и *b* начинается работа *c.*

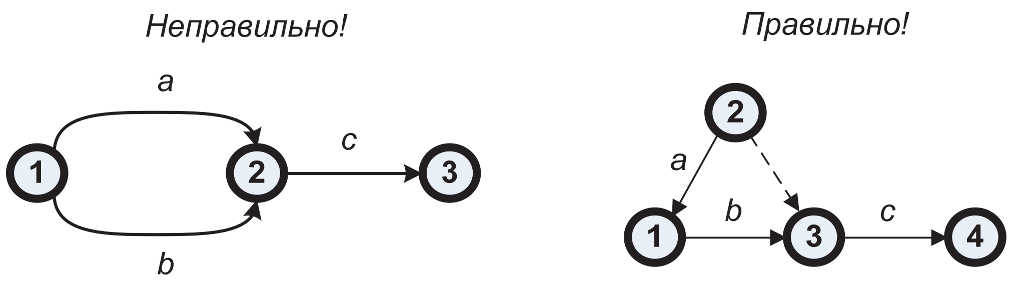


Рис. 1.41

На рис. 1.42 приведены неправильное и правильное изображения трех работ *a, b, c,* начинающихся после события 1; после их завершения начинается работа *d.*

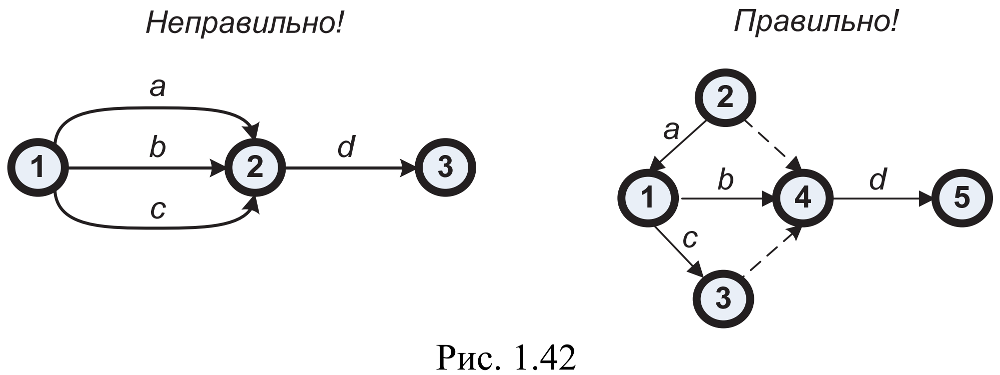


Рис. 1.42

Правило 6 важно для автоматических расчетов сетевых графиков, так как работы кодируются начальными и конечными событиями. При нарушении этого правила в памяти машины окажутся две или более одинаково закодированные работы.

Правило 7. Если какие-то работы могут начаться до полного завершения предыдущей работы, то ее следует разбить на части и считать каждую из них самостоятельной.

На рис. 1.43 изображена часть сетевого графика. Здесь работа a разбита на три части a_1, a_2 и a_3, причем после выполнения a_1 начинается работа b_1, после $a_2 – b_2$, после $a_3 – b_3$.

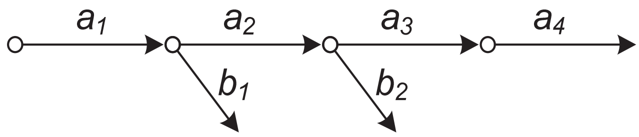


Рис. 1.43

Такая ситуация может возникнуть, например, при проведении труб (водопроводных или газовых). Укладку труб можно производить по частям, не дожидаясь, когда будет готова траншея на всем участке.

Правило 8. На сетевом графике следует четко отражать последовательность выполнения отдельных работ и их взаимосвязи. В помощь вводятся штриховые стрелки (условные зависимости) и дополнительные вершины (события).

Рассмотрим примеры построения отдельных фрагментов сетевых графиков.

Четыре работы (a, b, c, d) связаны между собой следующей зависимостью: b начинается после завершения работ a и c; d – после завершения работы c (рис. 1.44, *а*).

Пять работ (a, b, c, d, e) связаны между собой следующей зависимостью: c начинается после завершения a и b; e – после окончания b и d (рис. 1.44, *б*).

Пять работ (a, b, c, d, e) связаны между собой так: работы a и b начинаются после завершения каких-то одних и тех же работ; работу d можно начинать по окончании работ a и c; работу e – после завершения работ b и c (рис. 1.44, *в*).

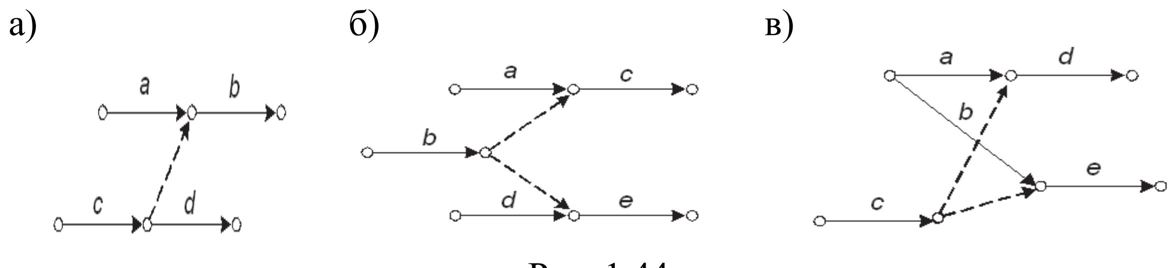


Рис. 1.44

1.5.3. Критический путь

*Определение 1.72. Время, необходимое для выполнения работы <i; j>, называют **продолжительностью работы** и обозначают t<i; j>. Обозначение проставляют над соответствующей стрелкой.*

На рис. 1.45 приведен сетевой график некоторого комплекса работ. Над стрелками проставлено время выполнения каждой из работ. Здесь $t < 0; 1 > = 10$; $t < 3; 1 > = 7$; $t < 4; 7 > = 10$.

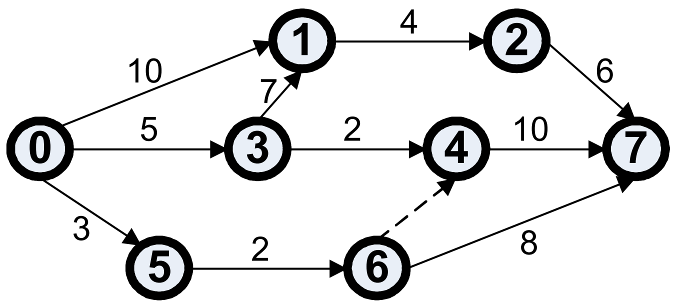


Рис. 1.45

Пусть сетевой график некоторого проекта (комплекса работ) построен. Время выполнения отдельных работ измерено, например, в неделях. Выясним, как по любому сетевому графику определить время, необходимое для реализации соответствующего проекта.

Заметим, что почти в любой сети от исходного события до завершающего ведет несколько путей. Каждому пути соответствует последовательность каких-то работ.

*Определение 1.73. Путь в сети от исходного события до завершающего называют **полным путем**. Полный путь будем обозначать L.*

*Определение 1.74. **Продолжительностью пути** в сетевом графике называют время, необходимое для выполнения всех работ, лежащих на этом пути.* Продолжительность полного пути L будем обозначать $t(L)$.

Пример 1.13

В сетевом графике на рис. 1.46 от исходного события 0 до события 7 ведут три пути. Обозначим их L_1, L_2, L_3. Пусть L_1 проходит через вершины 0, 1, 5, 7; L_2 – через вершины 0, 3, 7; L_3 – через вершины 0, 2, 4, 6, 7. Вычислим их продолжительность:

$t(L_1) = 3 + 4 + 2 = 9$;

$t(L_2) = 5 + 6 = 11$;

$t(L_3) = 1 + 2 + 2 + 3 = 8$.

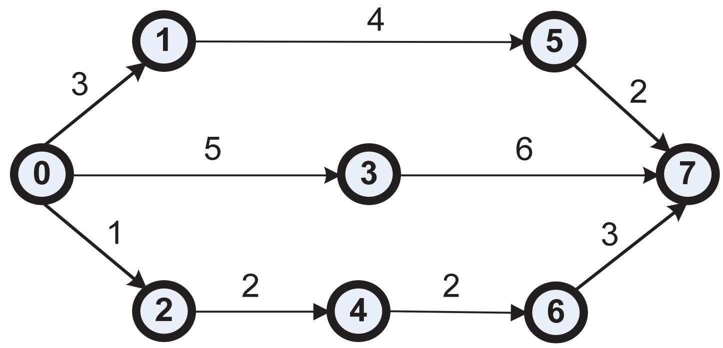


Рис. 1.46

Сравнив $t(L)$ для всех полных путей в сетевом графике на рис. 1.46, убеждаемся, что продолжительность самого «неблагоприятного» пути равна 11 неделям. Соответствующий проект не может быть реализован меньше, чем за 11 недель.

Определение 1.75. *Путь, имеющий наибольшую продолжительность, называется* ***критическим путем****. Критический путь будем обозначать* $L_{кр}$.

На рис. 1.46 критический путь проходит через вершины 0, 3 и 7.

Для определения времени, необходимого для реализации проекта, достаточно найти критический путь и вычислить его продолжительность. Продолжительность критического пути будем обозначать $t(L_{кр})$. Заметим, что в сети может быть несколько критических путей.

Определение 1.76. *Работы, лежащие на критическом пути, называются* ***критическими****. Работы, не лежащие на критическом пути, называются* ***некритическими***.

От продолжительности критических работ зависит общий срок завершения всего комплекса работ. Сокращение или увеличение сроков выполнения критических работ соответственно сокращает или увеличивает общую продолжительность выполнения проекта. На сетевом графике работы, лежащие на критическом пути, обозначают более жирными стрелками. Заметим, что направление штриховой стрелки влияет на продолжительность критического пути.

Некритические работы допускают некоторое запаздывание в их выполнении, которое не задержит сроков реализации всего проекта.

Введем определения отдельных **временных параметров** сетевого графика.

Определение 1.77. ***Ранее время начала работы*** *– это наиболее ранний из возможных сроков начала выполнения работы.*

Определение 1.78. ***Раннее время окончания работы*** *равно сумме раннего времени начала работы и продолжительности самой работы.*

Определение 1.79. ***Позднее время окончания работы*** *– это наиболее поздний из допустимых сроков окончания работы.*

Определение 1.80. ***Позднее время начала работы*** *равно разнице времени окончания работы и ее продолжительности.*

Определение 1.81. ***Наиболее ранний срок наступления события*** *характеризует наиболее ранний из возможных сроков свершения того или иного события.*

Поскольку каждое событие (кроме исходного) является результатом свершения одной или нескольких работ, а те, в свою очередь, следуют за какими-либо предшествующими событиями, то срок его наступления определяется величиной **наиболее длительного** отрезка пути от исходного события до рассматриваемого.

Для наиболее удобной компактной записи расчета временных параметров сетевого графика каждую вершину разделим на три сектора (рис. 1.47), в нижнем секторе будем проставлять номер события.

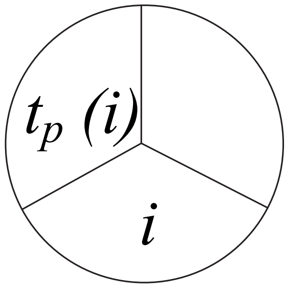


Рис. 1.47

Наиболее ранний из возможных сроков наступления события i обозначается $t_p(i)$. На сетевом графике t_p (i) будем проставлять в левом секторе (рис. 1.47).

Сформулируем **общее правило определения t_p (i)**.

1) Наиболее ранний срок наступления исходного события всегда равен 0, т.е. t_p $(0) = 0$.

2) t_p (i) равно продолжительности наиболее «неблагоприятного» (наибольшего по сумме всех предшествующих работ) пути, ведущего от исходного события к событию i.

Отметим, что вычисление наиболее ранних сроков наступления события осуществляется прямым ходом (в направлении от исходного события к завершающему). Для нахождения наиболее ранних сроков промежуточных и завершающего событий рассматривают все работы, непосредственно предшествующие данному событию. Для каждого из этих событий складывают продолжительность предшествующей работы и ранний срок наступления события, непосредственно предшествующего этой работе; сравнивают эти суммы, **наибольшая** из них определяет t_p (i):

$$t_{\text{р}}(i) = \max_j\{t_p(j) + t\langle j, i\rangle\},$$

где j – начальные события работ, непосредственно предшествующих событию i.

Пример 1.14

Найти наиболее ранние сроки наступления каждого из событий сетевого графика, представленного на рис. 1.48. Начнем последовательно рассматривать все вершины: $t_p(1) = 0 + 1$, так как только один путь ведет от исходного события к событию 1; к событию 2 ведут два ребра <1; 2>, <0; 2> и $t_p(2) = \max\{(1 + 3); 5\} = 5$. Это означает, что наступление события 2 нельзя ожидать раньше чем через 5 (недель); к событию 3 тоже ведут два ребра <1; 3>, <2; 3> и $t_p(3) = \max\{(1 + 2);\ (5+6)\} = 11$. Это означает, что наступления события 3 нельзя ожидать раньше чем через 11 (недель) (рис. 1.48). Аналогично подсчитываем, что

$t_p(4) = \max\{(5 + 5); 11\} = 11;$

$t_p(5) = \max\{(11 + 5); (11 + 3)\} = 16.$

Число 16 представляет собой время выполнения всего проекта (рис. 1.48).

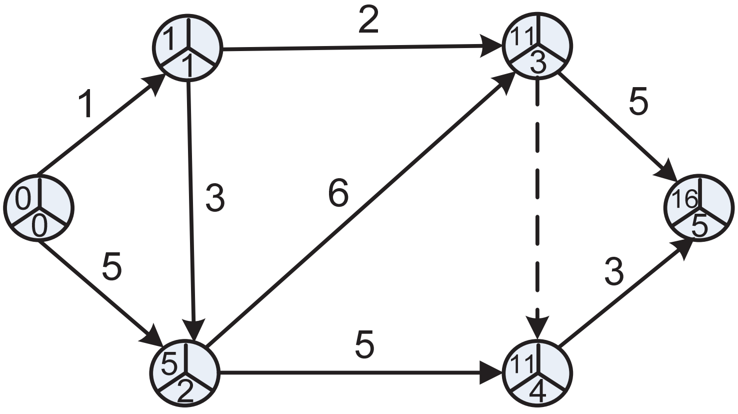


Рис. 1.48

Определение 1.82. ***Наиболее поздний срок наступления события*** *характеризует наиболее поздний из допустимых сроков совершения того или иного события.*

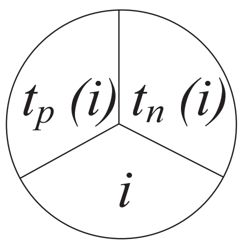


Рис. 1.49

Наиболее поздний срок наступления события – самый поздний срок наступления события, при котором планируемый срок окончания проекта не меняется.

Если установлен срок наступления завершающего события, являющегося результатом всего комплекса проводимых работ, то каждое промежуточное событие должно наступить не позже определенного срока.

Наиболее поздний срок наступления события i обозначается $t_n(i)$. На сетевом графике $t_n(i)$ будем проставлять в правом секторе (рис. 1.49).

Заметим, что для завершающего события k поздний срок наступления совпадает с ранним сроком наступления, то есть $t_p(k) = t_n(k)$.

При определении наиболее поздних сроков наступления событий расчет ведут от завершающего события к исходному (обратный ход вычислений).

Сформулируем **общее правило определения t_n *(i)***.

1) Определить поздний срок наступления завершающего события. Если завершающее событие обозначено j, то t_n (j) = t_p (j).

2) От завершающего события идем последовательно к исходному. Для всякого промежуточного события i определяем поздние сроки начала всех работ, начинающихся сразу после этого; находим среди них самый ранний, который равен t_n (i).

Пример 1.15

Определим поздние сроки наступления событий по сетевому графику на рис. 1.50. В правый сектор завершающего события 5 поставим t_n (5) = t_p (5) = 16.

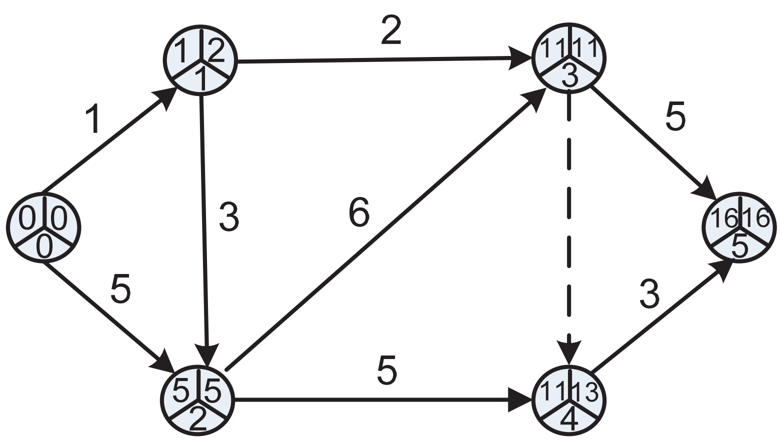


Рис. 1.50

Рассмотрим теперь событие 4; t < 4; 5 > = 3, так что работа <4; 5> может начаться не позднее, чем за 3 недели до события 5. Поздний срок наступления события 4 равняется 16 - 3 = 13. Поставим 13 в правый сектор соответствующей вершины на сетевом графике.

Событие 3 отделено от события 5 работой <3; 5> и от события 4 работой <3; 4>, причем t <3; 5> = 5, t<3; 4> = 0. Работа <3; 5> может начаться не позднее, чем за 5 недель до события 5, а работа <3; 4> – фиктивная, но она показывает зависимость событий 3 и 4; а именно поздний срок наступления события 3 не может быть больше, чем 13.

Таким образом, поздний срок наступления события 3 определяется наименьшей из разностей: 16 - 5 = 11 и 13 - 0 = 13. Следовательно, t_n (3) = = 11. (Ставим 11 в правый сектор.)

Событие 2 отделено от события 4 работой <2; 4> и от события 3 работой <2; 3>. Выбираем наименьшую из разностей: t_n (3) - t<2; 3> = 11 - 6 = = 5; t_n (*3*) - t<2; 4> = 13 - 5 = 8.

Следовательно, $t_n(2) = 5$. Событие 2 должно наступить не позднее, чем на 5-й неделе, так как иначе задержится общий срок завершения всего комплекса. (Ставим 5 в правый сектор.)

Аналогично определяем поздний срок наступления события 1:

$$t_n(1) = \min\{[t_n(3) - t<1; 3>]; [t_n(2) - t<1; 2>]\} =$$
$$= \min\{[11 - 2]; \ [5 - 3]\} = \min\{9; 2\} = 2.$$

Естественно, что поздний срок наступления исходного события 0 должен равняться 0.

Заметим, что для событий i, лежащих на критическом пути, ранние и поздние сроки наступлений совпадают, то есть $t_n(i) = t_p(i)$.

Некритические работы имеют некоторые резервы времени. Для некритических событий можно определить некоторый интервал времени, в течение которого наступление данного события не повлияет на время завершения всего комплекса, то есть резерв времени события.

При выполнении проекта важно бывает определить резервы времени какого-то события i. Это дает возможность узнать время, на которое можно увеличить срок выполнения какой-то из работ.

Определение 1.83. ***Резерв времени наступления события*** *– это разница между поздним и ранним сроками наступления этого события.*

Обозначим резерв времени события i через $r(i)$.

Существуют различные **алгоритмы для отыскания критического пути** и для определения его продолжительности. Мы познакомимся с одним из них.

Этапы нахождения критического пути:

1) находим наиболее ранние сроки наступления каждого события;
2) находим поздние сроки наступления каждого события;
3) вычисляем резервы времени, определяем критические события;
4) определяем критические работы и строим критический путь.

Пример 1.16

Определим резервы времени каждого из событий сетевого графика, представленного на рис. 1.51.

$r(0) = 0 - 0 = 0$; $r(1) = 2 - 1 = 1$; $r(2) = 5 - 5 = 0$;

$r(3) = 11 - 11 = 0$; $r(4) = 13 - 11 = 2$; $r(5) = 16 - 16 = 0$.

Таким образом, события, резервы которых равны нулю, являются критическими, это события 0, 2, 3, 5. Определяем критические работы и сам критический путь $L_{кр} = 16$ (рис. 1.51).

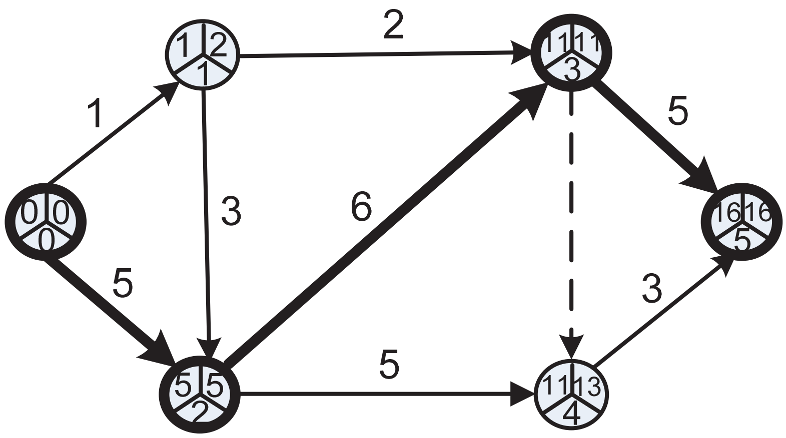


Рис. 1.51

Упражнения

1) Постройте фрагмент сетевого графика, если:

- начало работы *e* зависит только от окончания работ *a, b* и *c*; начало работы *d* – только от окончания работ *c* и *b*;
- начало работы *e* зависит только от окончания работ *a* и *c*; начало работы *d* – только от окончания работ *b* и *c*;
- работы *b* и *c* имеют началом одно и то же событие; начало работы *e* зависит только от окончания работ *a, b* и *c*; начало работы *d* – только от окончания работы *b*.

2) Постройте фрагмент сетевого графика, если:

- начало работы *e* зависит от окончания работы *c*; начало работы *d* – только от окончания работ *a* и *c*; начало работы *h* – только от окончания работ *b* и *e*;
- начало работы *e* зависит только от окончания работ *a* и *c*; начало работы *d* – только от окончания работы *c*; начало работы *h* – только от окончания работ *b* и *c*;
- начало работы *e* зависит от окончания работы *b*; начало работы *d* – от окончания работ *a, c* и *e*; начало работы *h* – от окончания работ *a* и *c*.

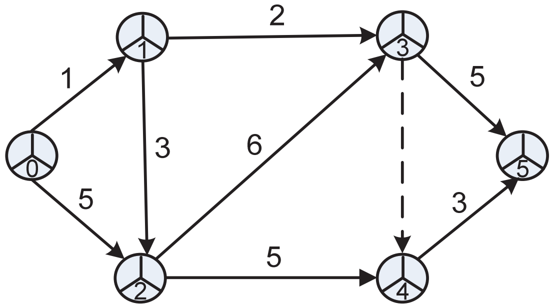
3) Постройте фрагменты сетевых графиков, если:

- работу *e* можно начать после окончания работ *a* и *c*; работу *h* – после окончания работ *b* и *c*;
- работы *a* и *c* имеют началом одно и то же событие; работа *e* может начаться после окончания работ *a* и *c*; работа *d* – после окончания работ *b* и с; работа *h* – после окончания работы *b*.

Проверочный тест по теме «Сетевое планирование и управление»

Вопрос 1.

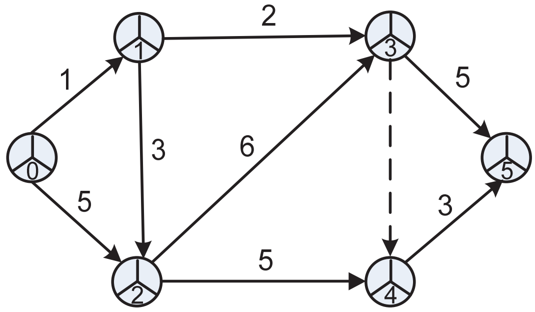
Для данного сетевого графика установите соответствие между номером события и его резервом времени.



1 Событие 1	а) $R = 2$
2 Событие 4	б) $R = 1$
3 Событие 3	в) $R = 0$

Вопрос 2.

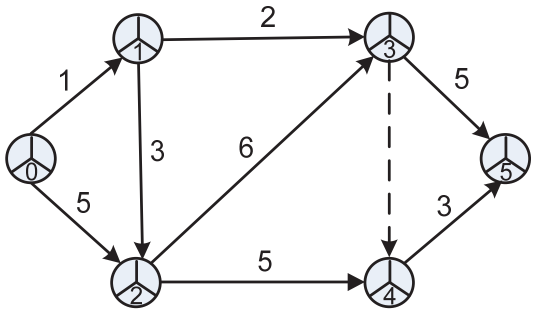
Для данного сетевого графика установите соответствие между номером события и его резервом времени.



1 Событие 1	а) $R = 2$
2 Событие 4	б) $R = 1$
3 Событие 2	в) $R = 0$

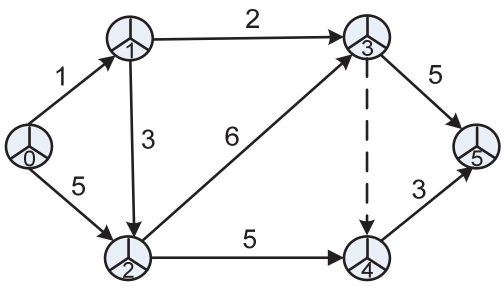
Вопрос 3.

По данному сетевому графику определите, чему равен резерв события 1.



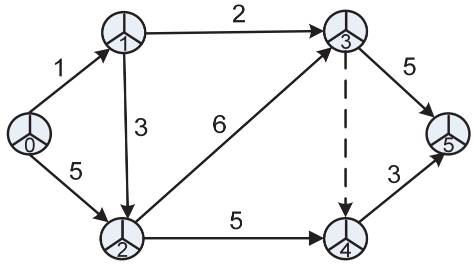
Вопрос 4.

По данному сетевому графику определите, чему равен резерв события 2.



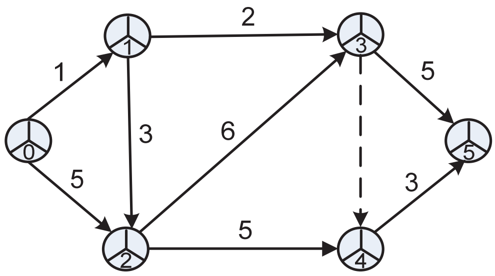
Вопрос 5.

По данному сетевому графику определите, чему равен резерв события 3.



Вопрос 6.

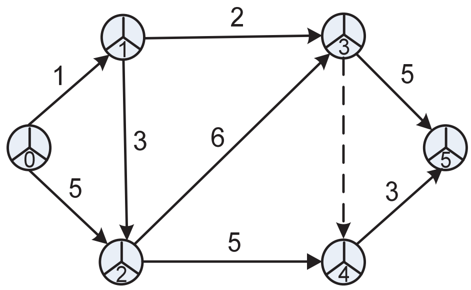
По данному сетевому графику определите, чему равен резерв события 4.



Вопрос 7.

По данному сетевому графику определите критический путь:

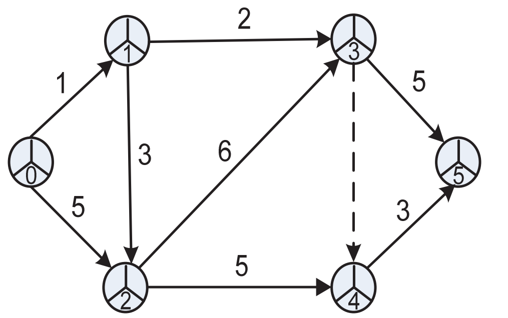
а) $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5$;
б) $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5$;
в) $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_5$;
г) $V_0 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5$.



Вопрос 8.

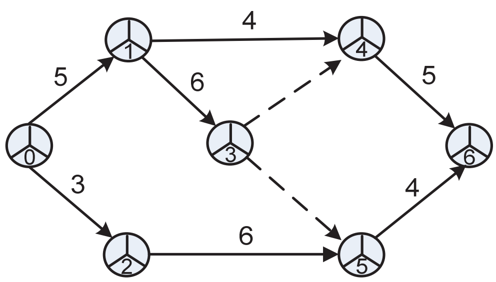
По данному сетевому графику определите порядок событий в критическом пути.

а) V_0;
б) V_3;
в) V_2;
г) V_5.



Вопрос 9.

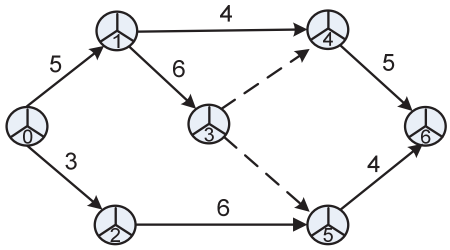
Для данного сетевого графика установите соответствие между номером события и его резервом времени.



1 Событие 1	а) $R = 0$
2 Событие 2	б) $R = 1$
3 Событие 5	в) $R = 3$

Вопрос 10.

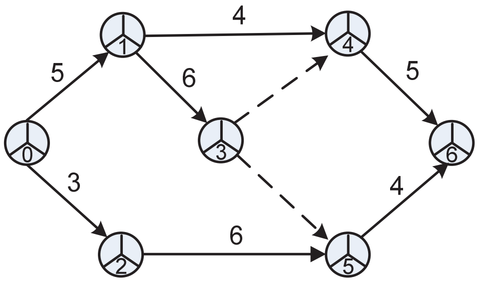
Для данного сетевого графика установите соответствие между номером события и его резервом времени.



1 Событие 3	а) $R = 0$
2 Событие 2	б) $R = 1$
3 Событие 5	в) $R = 3$

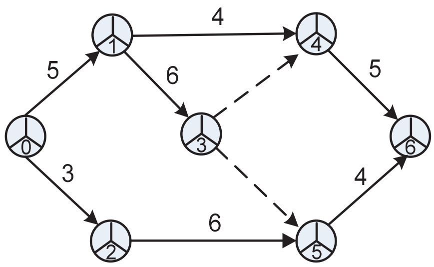
Вопрос 11.

По данному сетевому графику определите, чему равен резерв события 1.



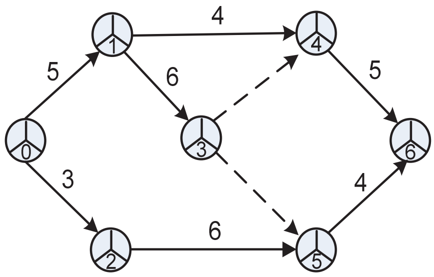
Вопрос 12.

По данному сетевому графику определите, чему равен резерв события 2.



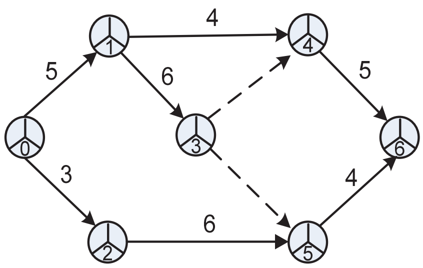
Вопрос 13.

По данному сетевому графику определите, чему равен резерв события 3.



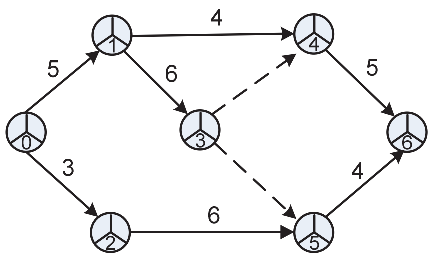
Вопрос 14.

По данному сетевому графику определите, чему равен резерв события 4



Вопрос 15.

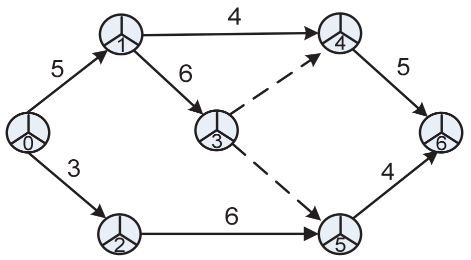
По данному сетевому графику определите, чему равен резерв события 5.



Вопрос 16.

По данному сетевому графику определите критический путь:

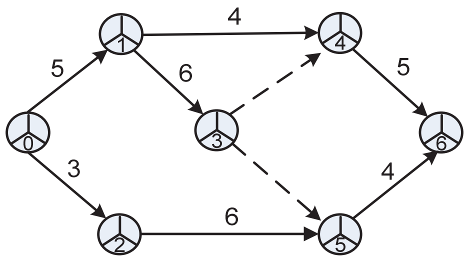
а) $V_0 \rightarrow V_1 \rightarrow V_4 \rightarrow V_6$;
б) $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_6$;
в) $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_6$;
г) $V_0 \rightarrow V_2 \rightarrow V_5 \rightarrow V_6$.



Вопрос 17.

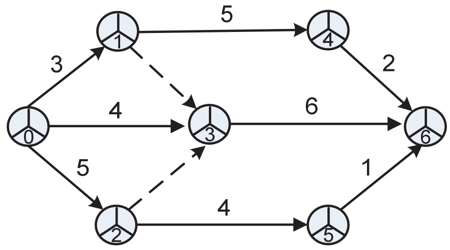
По данному сетевому графику определите порядок событий в критическом пути:

а) V_0; г) V_1;
б) V_3; д) V_6.
в) V_4;



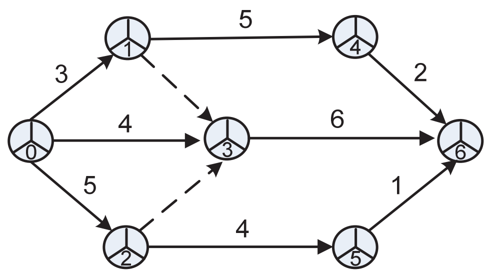
Вопрос 18.

По данному сетевому графику определите, чему равен резерв события 1.



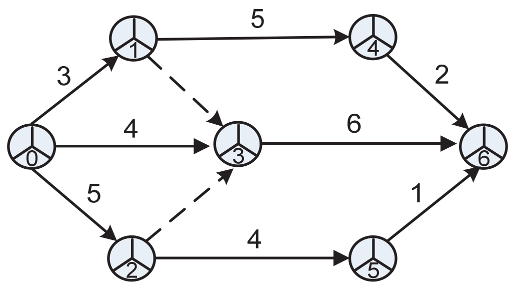
Вопрос 19.

По данному сетевому графику определите, чему равен резерв события 2.



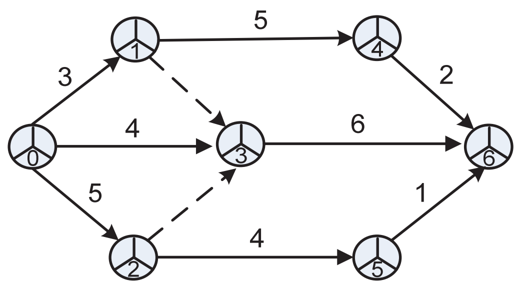
Вопрос 20.

По данному сетевому графику определите, чему равен резерв события 3.



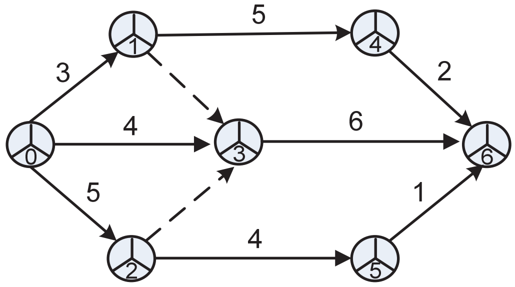
Вопрос 21.

По данному сетевому графику определите, чему равен резерв события 4.



Вопрос 22.

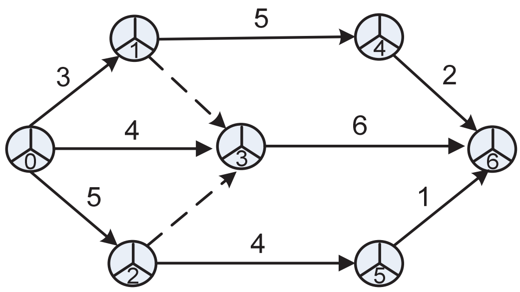
По данному сетевому графику определите, чему равен резерв события 5.



Вопрос 23.

По данному сетевому графику определите критический путь:

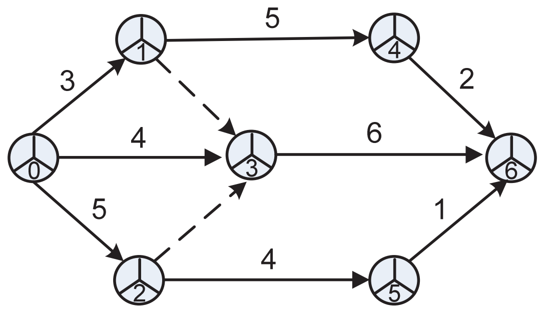
а) $V_0 \to V_2 \to V_3 \to V_6$;
б) $V_0 \to V_1 \to V_4 \to V_6$;
в) $V_0 \to V_2 \to V_5 \to V_6$;
г) $V_0 \to V_1 \to V_3 \to V_6$.



Вопрос 24.

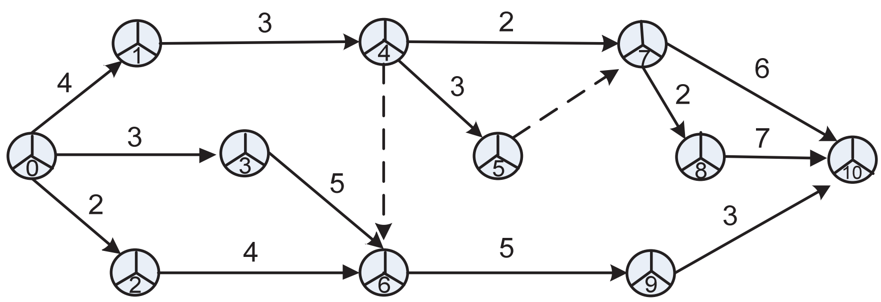
По данному сетевому графику определите порядок событий в критическом пути:

а) V_0;
б) V_3;
в) V_2;
г) V_6.



Вопрос 25.

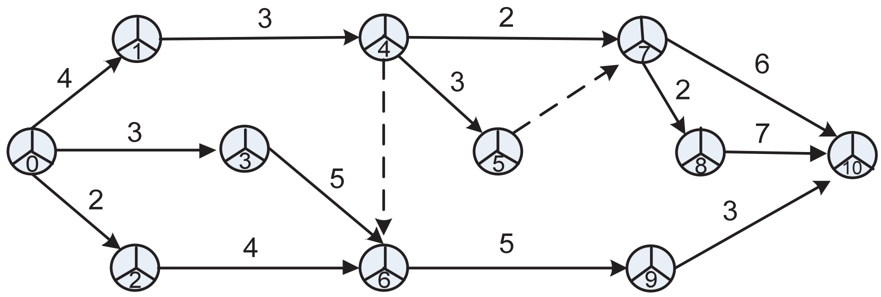
Для данного сетевого графика установите соответствие между номером события и его резервом времени.



1 Событие 9	а) $R = 5$
2 Событие 2	б) $R = 0$
3 Событие 5	в) $R = 3$

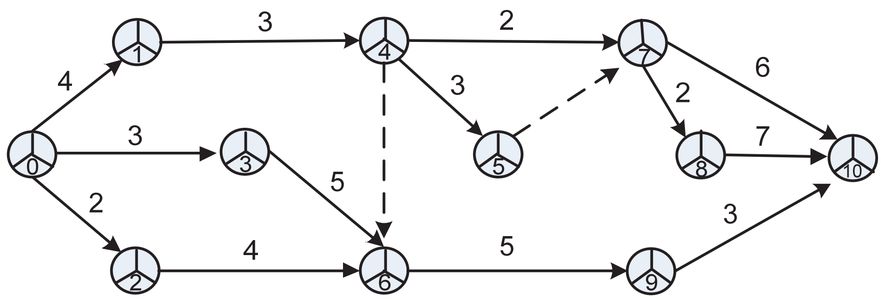
Вопрос 26.

Для данного сетевого графика установите соответствие между номером события и его резервом времени.

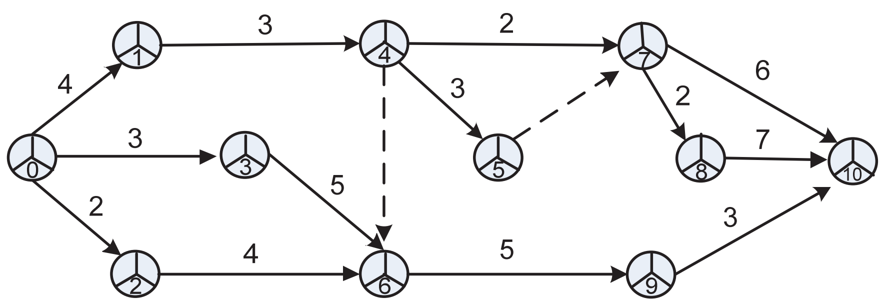


1 Событие 3	а) $R = 3$
2 Событие 4	б) $R = 5$
3 Событие 2	в) $R = 0$

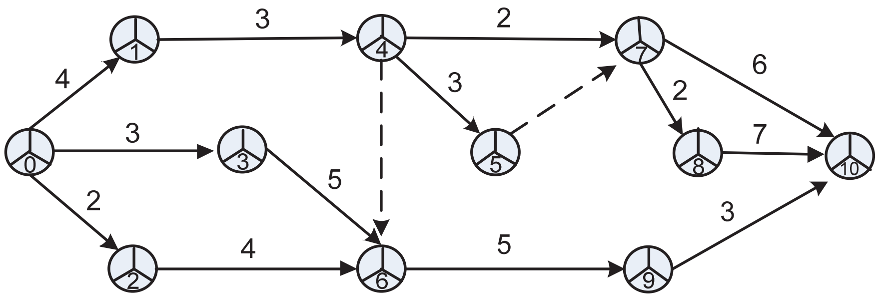
Вопрос 27. По данному сетевому графику определите, чему равен резерв события 1.



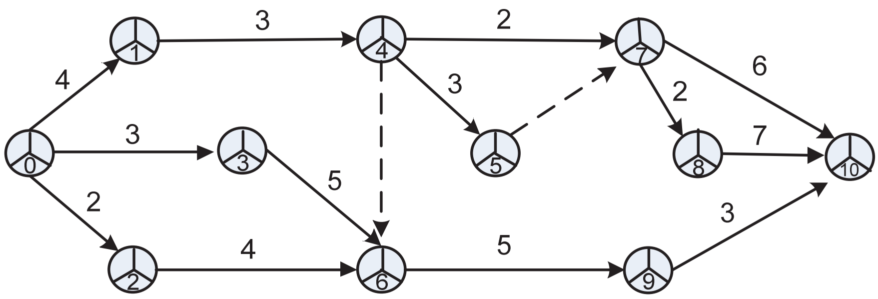
Вопрос 28. По данному сетевому графику определите, чему равен резерв события 2.



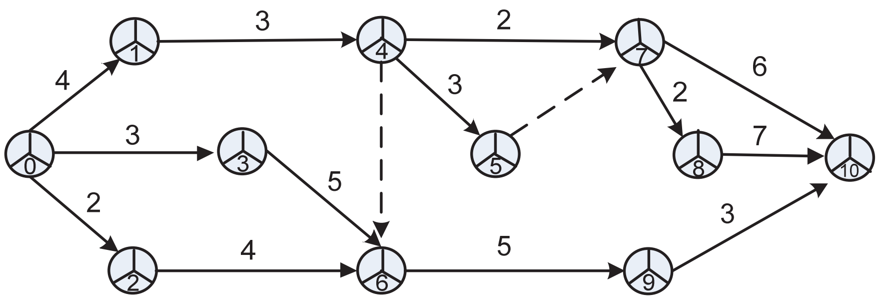
Вопрос 29. По данному сетевому графику определите, чему равен резерв события 3.



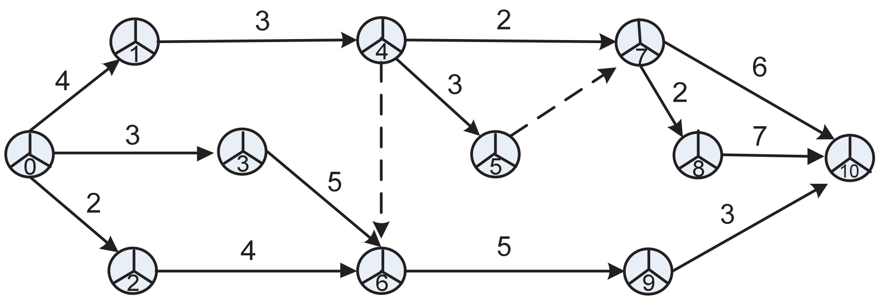
Вопрос 30. По данному сетевому графику определите, чему равен резерв события 4.



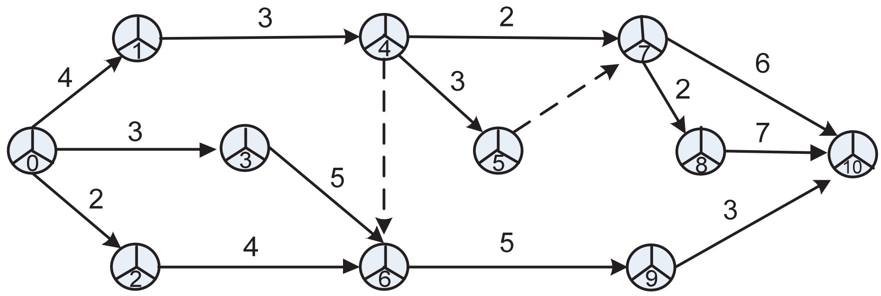
Вопрос 31. По данному сетевому графику определите, чему равен резерв события 5.



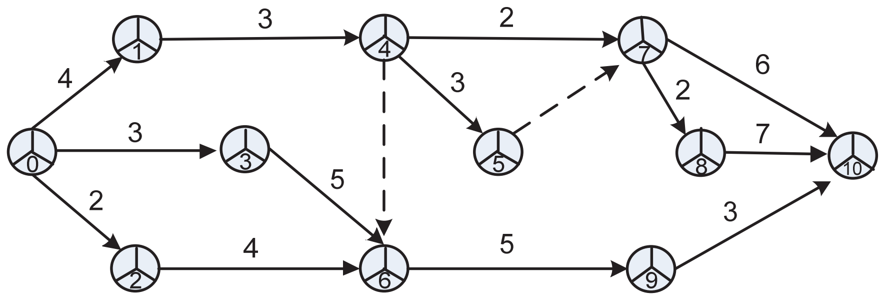
Вопрос 32. По данному сетевому графику определите, чему равен резерв события 6.



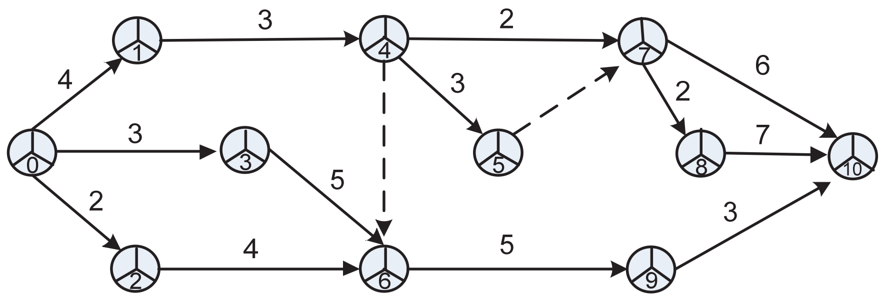
Вопрос 33. По данному сетевому графику определите, чему равен резерв события 7.



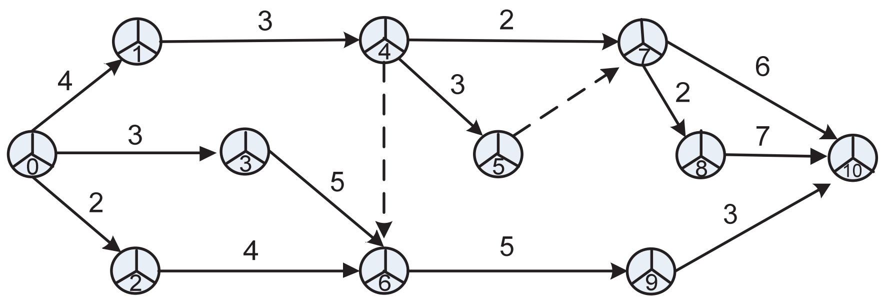
Вопрос 34. По данному сетевому графику определите, чему равен резерв события 8.



Вопрос 35. По данному сетевому графику определите, чему равен резерв события 9.

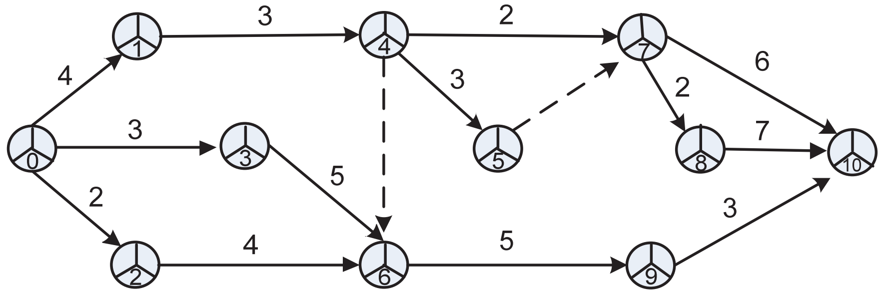


Вопрос 36. По данному сетевому графику определите критический путь.



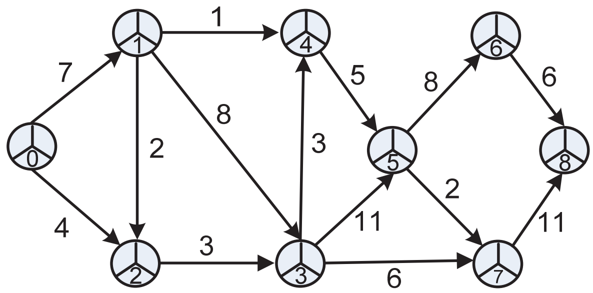
а) $V_0 \rightarrow V_1 \rightarrow V_4 \rightarrow V_5 \rightarrow V_7 \rightarrow V_{10}$;
б) $V_0 \rightarrow V_1 \rightarrow V_4 \rightarrow V_5 \rightarrow V_7 \rightarrow V_8 \rightarrow V_{10}$;
в) $V_0 \rightarrow V_3 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{10}$;
г) $V_0 \rightarrow V_1 \rightarrow V_4 \rightarrow V_6 \rightarrow V_9 \rightarrow V_{10}$.

Вопрос 37. По данному сетевому графику определите порядок событий в критическом пути.



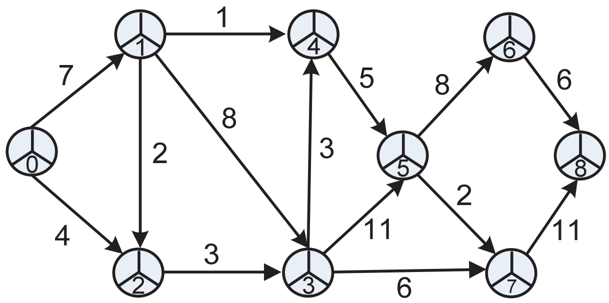
а) V_0;
б) V_8;
в) V_4;
г) V_1;
д) V_5;
е) V_7;
ж) V_{10}.

Вопрос 38. Для данного сетевого графика установите соответствие между номером события и его резервом времени.

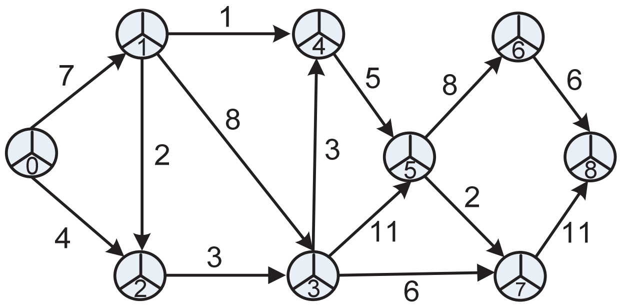


1 Событие 6	а) $R = 1$
2 Событие 7	б) $R = 3$
3 Событие 2	в) $R = 0$

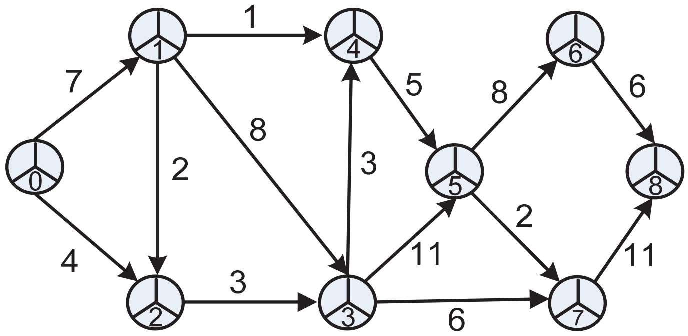
Вопрос 39. По данному сетевому графику определите, чему равен резерв события 1.



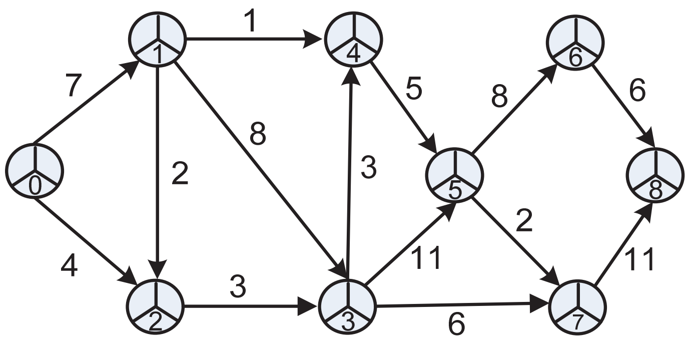
Вопрос 40. По данному сетевому графику определите, чему равен резерв события 2.



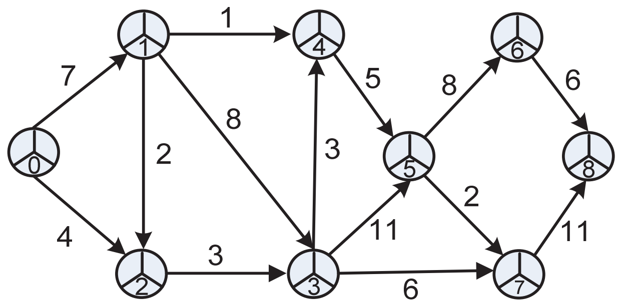
Вопрос 41. По данному сетевому графику определите, чему равен резерв события 3.



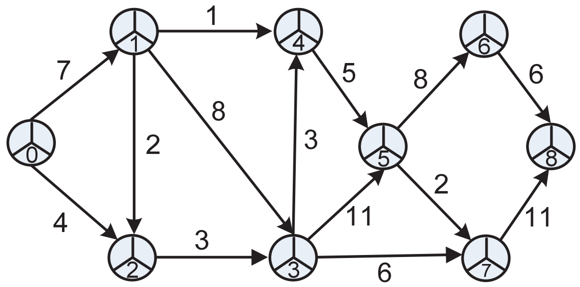
Вопрос 42. По данному сетевому графику определите, чему равен резерв события 4.



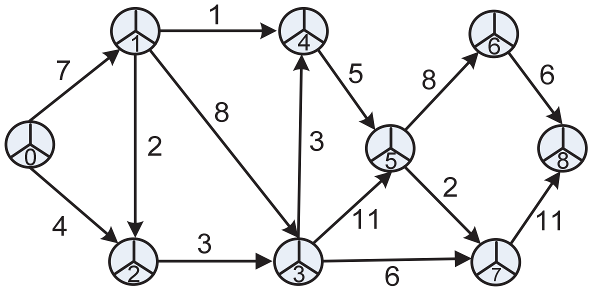
Вопрос 43. По данному сетевому графику определите, чему равен резерв события 5.



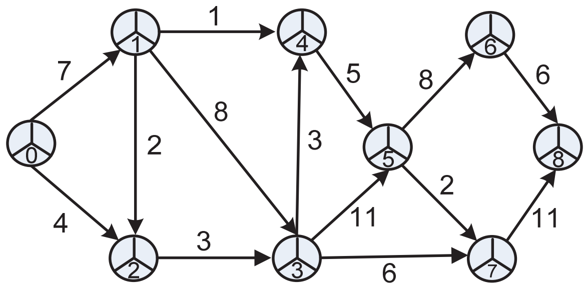
Вопрос 44. По данному сетевому графику определите, чему равен резерв события 6.



Вопрос 45. По данному сетевому графику определите, чему равен резерв события 7.

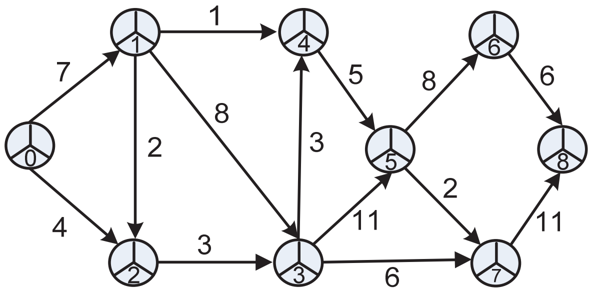


Вопрос 46. По данному сетевому графику определите критический путь.



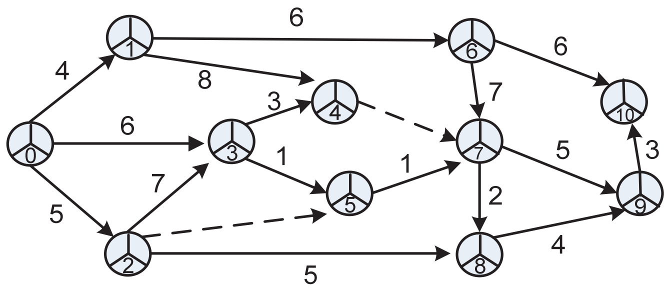
а) $V_0 \to V_2 \to V_3 \to V_7 \to V_8$;
б) $V_0 \to V_2 \to V_3 \to V_5 \to V_6 \to V_8$;
в) $V_0 \to V_1 \to V_2 \to V_3 \to V_5 \to V_6 \to V_8$;
г) $V_0 \to V_1 \to V_3 \to V_5 \to V_6 \to V_8$.

Вопрос 47. По данному сетевому графику определите порядок событий в критическом пути:



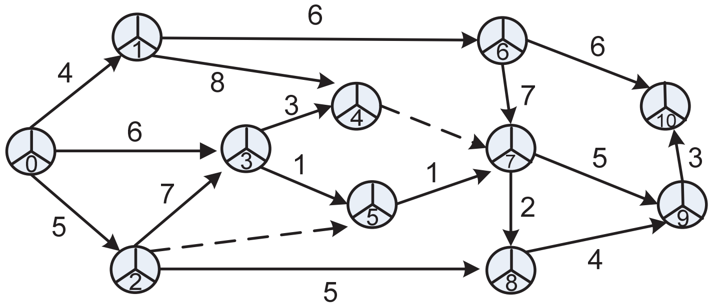
а) V_0; г) V_1;
б) V_8; д) V_5;
в) V_3; е) V_6.

Вопрос 48. По данному сетевому графику определите критический путь.



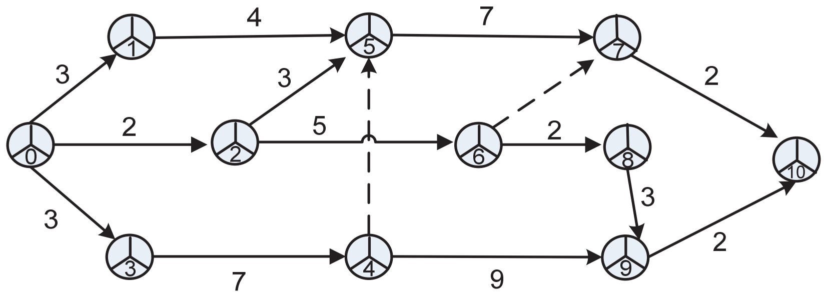
а) $V_0 \to V_1 \to V_6 \to V_7 \to V_8 \to V_9 \to V_{10}$;
б) $V_0 \to V_3 \to V_4 \to V_7 \to V_8 \to V_9 \to V_{10}$;
в) $V_0 \to V_2 \to V_5 \to V_7 \to V_8 \to V_9 \to V_{10}$;
г) $V_0 \to V_1 \to V_6 \to V_{10}$.

Вопрос 49. По данному сетевому графику определите порядок событий в критическом пути:



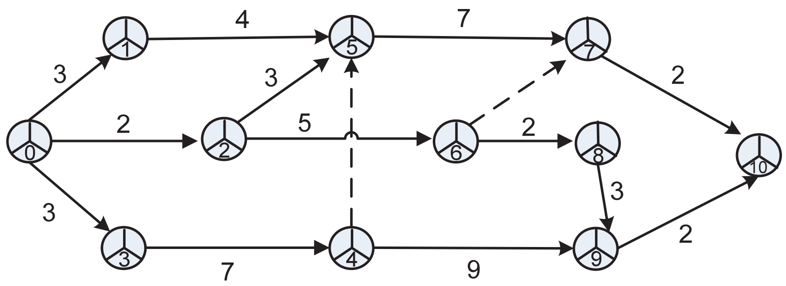
а) V_6; д) V_8;
б) V_0; е) V_7;
в) V_9; ж) V_{10}.
г) V_1;

Вопрос 50. По данному сетевому графику определите критический путь.



а) $V_0 \rightarrow V_1 \rightarrow V_5 \rightarrow V_7 \rightarrow V_{10}$;
б) $V_0 \rightarrow V_3 \rightarrow V_4 \rightarrow V_9 \rightarrow V_{10}$;
в) $V_0 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_7 \rightarrow V_{10}$;
г) $V_0 \rightarrow V_2 \rightarrow V_6 \rightarrow V_8 \rightarrow V_9 \rightarrow V_{10}$.

Вопрос 51. По данному сетевому графику определите порядок событий в критическом пути.



а) V_0;
б) V_4;
в) V_9;
г) V_3;
д) V_{10}.

2. ТЕОРИЯ АВТОМАТОВ

2.1. Конечные автоматы

Теория автоматов занимается изучением процессов, протекающих в автоматах различного рода, и общих закономерностей, которым они подчинены, широко применяя для этого алгебраический аппарат, математическую логику, комбинаторный анализ и теорию вероятностей. Конструируя надежные, хорошо работающие автоматы, приходится решать необычайно сложные задачи. Например, надо определять устойчивость систем, чтобы уменьшить различные отклонения в работе автоматических машин. Надо изучать и чувствительность автоматов, так как в процессе работы свойства систем регулирования не остаются постоянными.

Теория автоматов – это раздел теории управляющих систем, изучающий математические модели преобразователей дискретной информации, называемые автоматами. С определенной точки зрения такими преобразователями являются как реальные устройства (вычислительные машины, живые организмы), так и абстрактные системы (например, ***формальная система*** – это совокупность абстрактных объектов, не связанных с внешним миром, в котором представлены правила оперирования множеством символов в строго синтаксической трактовке без учета смыслового содержания, т.е. семантики; ***аксиоматические теории***, описывающие определенную совокупность явлений в их причинно-следственной связи друг с другом).

Наиболее тесно теория автоматов связана с ***теорией алгоритмов.*** Это объясняется тем, что автомат преобразует дискретную информацию по шагам в дискретные моменты времени и формирует результирующую информацию по шагам заданного алгоритма. Эти преобразования возможны с помощью технических и/или программных средств. Автомат можно представить как некоторое устройство (чёрный ящик), на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния. При анализе автоматов изучают их поведение при различных возмущающих воздействиях и минимизируют число состояний автомата для работы по заданному алгоритму. Такой автомат называют ***абстрактным***, так как абстрагируются от реальных физических входных и выходных сигналов, рассматривая их просто как буквы некоторого алфавита и в связи с идеализированным дискретным временем. Особое место в теории автоматов занимает понятие конечного автомата.

Конечный автомат является одним из важнейших видов управляющих систем. Главным достоинством конечных автоматов является то, что в них естественным образом описываются системы, управляемые внешними событиями.

Эти устройства имеют конечное число входов, воспринимающих информацию, и конечное число выходов для выдачи переработанной информации. Зависимость между входами и выходами задается предписанным алгоритмом переработки информации. Информация на входе и выходе представляется символами, физическими носителями которых являются квантованные по времени сигналы. Автоматы обладают следующими общими свойствами:

1) Всякая машина состоит из конечного множества элементов, каждый из которых в любой данный момент времени может находиться лишь в одном из конечного числа устойчивых состояний. Поэтому и вся машина имеет лишь конечное множество устойчивых состояний.

2) Каждая машина работает последовательно. Состояния машины меняются в четкой последовательности.

3) Машина является детерминированным устройством: при наличии полной информации о внутренних состояниях всех элементов машины и всех ее входов следующее состояние машины определено однозначно.

С функциональной точки зрения современная универсальная машина состоит из пяти типов устройств:

1) устройства ввода;
2) памяти;
3) арифметического устройства;
4) устройства управления;
5) устройства вывода.

Устройства ввода служат для ввода данных и команд. Они могут также измерять те или иные физические величины и переводить результаты в двоичную систему, которую машина способна воспринять. Устройства вывода служат для передачи информации от машины к пользователям.

Результат преобразования «***вход → выход***» зачастую зависит не только от входа в данный момент времени, но и от того, что было раньше на входе, от входной истории, т.е. от предыстории преобразования. Число возможных входных историй бесконечно (счетно), даже если различных элементов входной информации у автомата конечное число (как в конечном функциональном преобразователе). Чтобы эти предыстории как-то запоминать и отличать друг от друга, преобразователь должен иметь память. Чтобы машина выполнила необходимую последовательность команд, в ее память, прежде всего, должна быть заложена программа, предопределяющая эту последовательность.

Машина выполняет эту программу последовательно, команду за командой, начиная с первой. Однако сама программа может предусматривать пропуск или повторение тех или иных шагов, и эти пропуски и повторения управляемы; они, в принципе, зависят от характера информации, хранящейся в памяти.

Арифметическое устройство машины совершает как арифметические, так и логические операции.

Устройство управления интерпретирует команды, считываемые из памяти, вызывая их по порядку или иногда переставляя команды в программе в зависимости от характера данных, запасенных в памяти или вычисленных ранее.

Введем теперь математическое понятие, являющееся абстракцией описанных выше представлений.

Определение 2.1. Конечным автоматом называется набор из пяти объектов $[A, S, Z, \nu, \xi]$.

Здесь $A = \{a_0, a_1, \ldots, a_n\}$ *– конечный список входных символов (входной алфавит);*

$Z = \{z_0, z_1, \ldots, z_m\}$ *– список выходных символов (выходной алфавит);*

$S = \{s_0, s_1, ..., s_r\}$ – множество внутренних состояний;

ν: $S \times A \to S$ – функция перехода (в следующее состояние);

ξ: $S \times A \to Z$ – функция выхода.

Тем самым, конечный автомат математически описывается тремя множествами и двумя функциями. Действие его состоит в том, что он «считывает» последовательность входных символов и затем «выпечатывает» последовательность выходных символов. Действие происходит последовательно. Конечный автомат, находящийся сначала во внутреннем состоянии s_j, считывает первый входной символ a_k. Функция ξ принимает на паре (s_j, a_k) значение z_l, которое выпечатывается в качестве первого выходного символа. Функция ν принимает на паре (s_j, a_k) значение s_r, которое является следующим внутренним состоянием автомата. Затем автомат считывает новый входной символ, выпечатывает выходной, переходит в следующее состояние и т. д., пока не кончится программа.

На рис. 2.1 дан удобный способ представления последовательных тактов работы автомата.

Будем считать, что программа записана на входной ленте. Автомат считывает с нее входные символы один за другим. По прочтении каждого входного символа выпечатывается выходной символ на выходной ленте, и автомат переходит в следующее состояние, прежде, чем считать следующий символ программы.

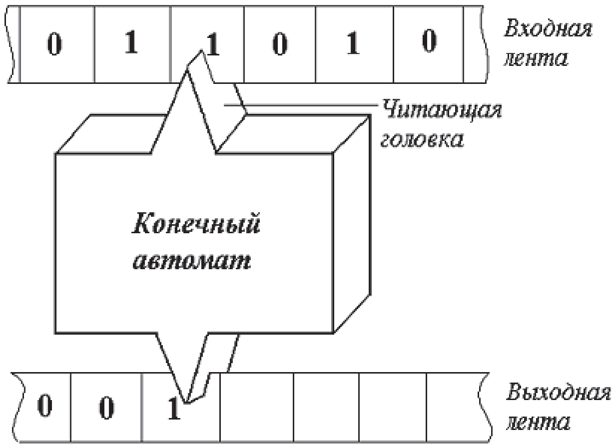


Рис. 2.1

В нашем определении подразумевается, что функции ν и ξ в описании автомата M всюду определены: каждый элемент $S \times A$ задает их значения. Такое описание автомата является полным. Коль скоро задано начальное состояние такого автомата, он способен считывать любую программу и выдавать однозначно определенную цепочку символов. Иными словами, существует функция, которая ставит в соответствие любому начальному состоянию s_i и любой последовательности входных символов вполне определенную последовательность выходных символов.

Пример 2.1

Рассмотрим следующий конкретный автомат $M = [A, S, Z, \nu, \xi]$. Входной алфавит $A = \{0, 1\}$; выходной алфавит $Z = \{0, 1\}$; три внутренних состояния $S = \{s_0, s_1, s_2\}$; функции выхода и перехода задаются предписаниями:

ν:	$(s_0,0)\mapsto s_1$ $(s_0,1)\mapsto s_0$ $(s_1,0)\mapsto s_2$ $(s_1,1)\mapsto s_1$ $(s_2,0)\mapsto s_0$ $(s_2,1)\mapsto s_2$	ξ:	$(s_0,0)\mapsto 0$ $(s_0,1)\mapsto 1$ $(s_1,0)\mapsto 1$ $(s_1,1)\mapsto 0$ $(s_2,0)\mapsto 1$ $(s_2,1)\mapsto 0$

Подадим на вход последовательность 0, 1, 0, 1. Если автомат находился сначала в состоянии s_0, то, считав первый символ 0, он перейдет сначала в состояние s_1 и выпечатает 0. Считав затем 1, он останется в состоянии s_1 и выпечатает 0. Считав следующий 0, он перейдет в состояние s_2 и выпечатает 1. Наконец, считав последний символ 1, автомат закончит работу в состоянии s_2, имея на выходной ленте последовательность 0, 0, 1, 0. Таким образом, автомат преобразовал вход 0101 в 0010.

Есть два удобных способа описать этот автомат.

Прежде всего, можно построить помеченный ориентированный граф, называемый ***диаграммой состояний***.

Вершины графа помечены символами, обозначающими внутренние состояния. Каждое ребро помечено парой символов *a, z*, где *a* – входной символ, вызывающий переход в следующее состояние, отвечающее этому ребру, а *z* – выходной символ, который автомат выпечатывает.

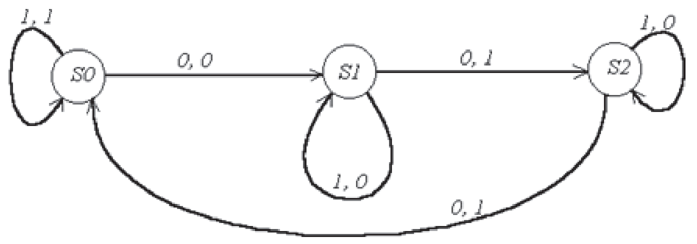


Рис. 2.2

Например, на рис. 2.2 изображен граф (диаграмма состояний) для автомата из примера 2.1.

Второй способ описания – таблица состояний. Это просто табличное представление функций ν, ξ. Например, таблица состояний для автомата из примера 2.1 будет выглядеть следующим образом (табл. 2.1).

Таблица 2.1

Текущее состояние	Следующее состояние ν		Выходной символ ξ	
	0	1	0	1
S_0	S_1	S_0	0	1
S_1	S_2	S_1	1	0
S_2	S_0	S_2	1	0

Оба способа имеют свои преимущества и недостатки. Таблица обычно удобнее при вычислениях, диаграмма нагляднее. Например, по диаграмме легко обнаружить состояния, не достижимые из других состояний.

Задача 2.1

Начертить диаграмму состояний автомата с алфавитами $A = Z = \{0, 1\}$, который выпечатывает 1, если непосредственно перед этим он считал четыре последовательных 1; в противном случае выпечатывает 0. Автомат работает так до тех пор, пока не считает три последовательных 0, после чего выпечатывает лишь нули.

Решение.

Входной и выходной алфавиты совпадают $A = Z = \{0, 1\}$. Рассмотрим внутренние состояния автомата, функции перехода и выхода. Описывая внутренние состояния автомата, помним, что он способен запоминать (добавлять в память) только по одному символу.

S_0 – начальное состояние,	ν: $(S_0, 0) \mapsto s_1$,	ξ: $(S_0, 0) \mapsto 0$,
S_1 – «0»,	$(S_0, 1) \mapsto s_2$,	$(S_0, 1) \mapsto 0$,
S_2 – «1»,	$(S_1, 0) \mapsto s_3$,	$(S_1, 0) \mapsto 0$,
S_3 – «00»,	$(S_1, 1) \mapsto s_2$,	$(S_1, 1) \mapsto 0$,
S_4 – «11»,	$(S_2, 0) \mapsto s_1$,	$(S_2, 0) \mapsto 0$,
S_5 – «000»,	$(S_2, 1) \mapsto s_4$,	$(S_2, 1) \mapsto 0$,
S_6 – «111»,	$(S_3 , 0) \mapsto s_5$,	$(S_3 , 0) \mapsto 0$,
	$(S_3 , 1) \mapsto s_2$,	$(S_3 , 1) \mapsto 0$,
	$(S_4 , 0) \mapsto s_1$,	$(S_4 , 0) \mapsto 0$,
	$(S_4 , 1) \mapsto s_6$,	$(S_4 , 1) \mapsto 0$,
	$(S_5 , 0) \mapsto s_5$,	$(S_5 , 0) \mapsto 0$,
	$(S_5 , 1) \mapsto s_5$,	$(S_5 , 1) \mapsto 0$,
	$(S_6 , 0) \mapsto s_1$,	$(S_6 , 0) \mapsto 0$,
	$(S_6 , 1) \mapsto s_6$,	$(S_6 , 1) \mapsto 1$.

Особое внимание следует обратить на состояние S_5. Обязательное наличие такого состояния обеспечит условие: «автомат работает так до тех пор, пока не считает три последовательных 0, после чего выпечатывает лишь нули». Для того чтобы обеспечить выполнение условия: «автомат выпечатывает 1, если непосредственно перед этим он считал четыре последовательных 1», достаточно состояния S_6 и условия, что, находясь в этом внутреннем состоянии, автомат считает еще одну единицу.

Построим таблицу (табл. 2.2) и диаграмму (рис. 2.3) состояний автомата.

Таблица 2.2

Текущее состояние	Следующее состояние ν		Выходной символ ξ	
	0	1	0	1
S_0	S_1	S_2	0	0
S_1	S_3	S_2	0	0
S_2	S_1	S_4	0	0
S_3	S_5	S_2	0	0
S_4	S_1	S_6	0	0
S_5	S_5	S_5	0	0
S_6	S_1	S_6	0	1

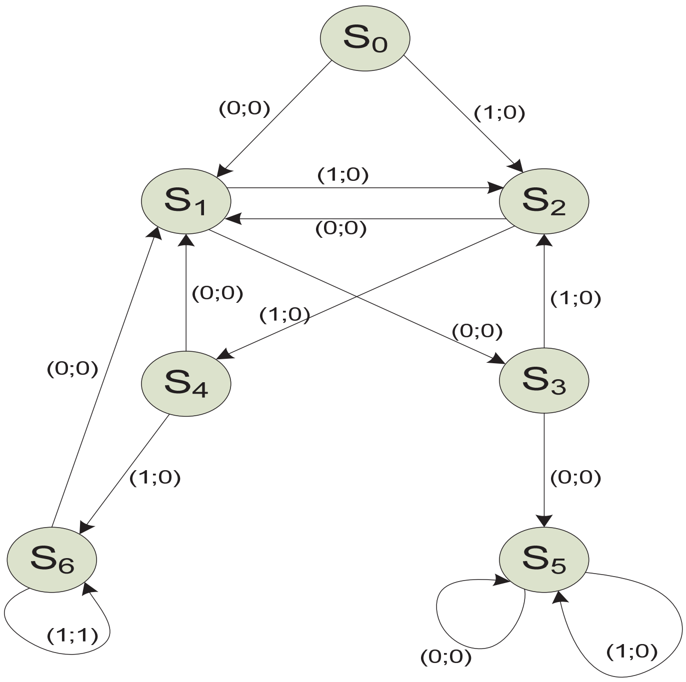


Рис. 2.3

Анализируя диаграмму состояний автомата, можно заметить следующее:

- в состоянии S_0 автомат может побывать только один раз, причем в самом начале своей работы;
- состояние S_5 достижимо только из состояния S_3;
- состояние S_5 является «зацикленным», т. е. автомат, попав в него, фактически перестает реагировать на входные данные и печатает только нули.

Для проверки работы построенного автомата зададим произвольную входную последовательность (например, «011111000110») и определим по таблице состояний (см. табл. 2.2), какую выходную последовательность получим в результате. Помним, что цикл работы автомата состоит из следующей последовательности шагов: «считал – напечатал – перешел в новое состояние» (табл. 2.3).

Таблица 2.3

Состояние	S_0	S_1	S_2	S_4	S_6	S_6	S_6	S_1	S_3	S_5	S_5	S_5
Вход	0	1	1	1	1	1	0	0	0	1	1	0
Выход	0	0	0	0	1	1	0	0	0	0	0	0

Задача 2.2

Описать автомат M с алфавитами $A = Z = \{0, 1\}$, который, исходя из начального состояния s_0, перерабатывает любую входную последовательность a^0a^1... в последовательность $00a^0a^1a^3$... (т. е. выход совпадает со входом, задержанным на два такта).

Решение.

Для простоты описания автомата предлагаем рассмотреть один из частных случаев. Пусть на вход автомата подана входная последовательность «011010», тогда на выходе должна быть получена следующая выходная последовательность «000110». Представим это в виде таблицы (табл. 2.4).

Таблица 2.4

Вход	0	1	1	0	1	0
Выход	0	0	0	1	1	0

Стрелки в таблице указывают, что выходной символ совпадает с входным символом, задержанным на два такта.

Входной и выходной алфавиты совпадают $A = Z = \{0, 1\}$.

Независимо от того, как будет выглядеть входная последовательность, два первых символа выходной последовательности будут нули. Поскольку входной алфавит состоит только из нулей и единиц, то возможны всего четыре варианта комбинаций, которые необходимо запомнить автомату: «00», «01», «10», «11». Исходя из этого, определяем внутренние состояния автомата:

S_0 – начальное состояние (память пуста),

S_1 – «0»,

S_2 – «1»,

S_3 – «00»,

S_4 – «01»,

S_5 – «10»,

S_6 – «11».

Рассуждая логически, определяем функцию перехода в новое состояние и функцию выхода. Представим автомат в виде таблицы состояний (табл. 2.5).

Таблица 2.5

Текущее состояние	Следующее состояние ν		Выходной символ ξ	
	0	1	0	1
S_0	S_1	S_2	0	0
S_1	S_3	S_4	0	0
S_2	S_5	S_6	0	0
S_3	S_3	S_4	0	0
S_4	S_5	S_6	0	0
S_5	S_3	S_4	1	1
S_6	S_5	S_6	1	1

Построим для автомата диаграмму состояний (рис. 2.4).

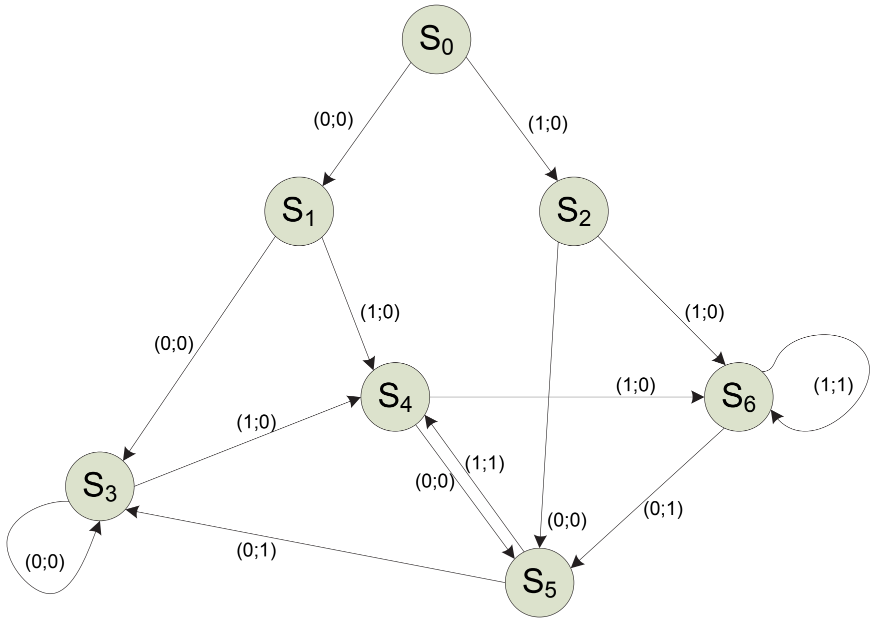


Рис. 2.4

Предлагаем самостоятельно проанализировать диаграмму состояний автомата.

Для проверки работы построенного автомата зададим произвольную входную последовательность (например, «011111000110») и определим по таблице состояний (см. табл. 2.5), какую выходную последовательность получим в результате. Помним, что цикл работы автомата состоит из следующей последовательности шагов: «считал – напечатал – перешел в новое состояние» (табл. 2.6).

Таблица 2.6

Состояние	S_0	S_1	S_4	S_6	S_6	S_6	S_6	S_5	S_3	S_3	S_4	S_6
Вход	0	1	1	1	1	1	0	0	0	1	1	0
Выход	0	0	0	1	1	1	1	1	0	0	0	1

Задача 2.3

Построить таблицу состояний автомата, у которого входной алфавит $A = \{a, b, c\}$, выходной алфавит $Z = (0; 1; 2; 3; 4)$ и автомат печатает 0 – если непосредственно перед этим считал последовательность «*abc*», печатает 1 – если непосредственно перед этим считал последовательность «*bca*», печатает 2 – если непосредственно перед этим считал последовательность «*cba*». Если автомат считал последовательно два одинаковых символа («*aa*», «*bb*», «*cc*»), то в этих случаях он печатает 3, в остальных случаях автомат печатает 4.

Решение.

Входной алфавит $A = \{a, b, c\}$, выходной алфавит $Z = \{0; 1; 2; 3; 4\}$. Зададим внутренние состояния автомата, помня о том, что автомат считывает и добавляет в память только по одному символу.

Запишем условия работы автомата в виде таблицы (табл. 2.7).

Таблица 2.7

Номер условия	Входная последовательность	Реакция автомата (выходной символ)
1	*abc*	0
2	*bca*	1
3	*cba*	2
4	*aa*	3
5	*bb*	3
6	*cc*	3
7	Остальные случаи	4

Чтобы обеспечить выполнение условий 1 – 3 в памяти автомата достаточно запомнить два первых символа, третий будет считан, поэтому вводим внутренние состояния:

S_0 – начальное состояние (память пуста),

S_1 – «*a*»,

S_2 – «*ab*»,

S_3 – «*b*»,

S_4 – «*bc*»,

S_5 – «*c*»,

S_6 – «*cb*».

Заметим, что для выполнения условий 4 – 6 достаточно наличие внутренних состояний S_1, S_3, S_5 соответственно.

Итак, составим таблицу состояний автомата (табл. 2.8).

Таблица 2.8

Текущее состояние	Следующее состояние ν			Выходной символ ξ		
	a	*b*	*c*	*a*	*b*	*c*
S_0	S_1	S_3	S_5	4	4	4
S_1	S_1	S_2	S_5	3	4	4
S_2	S_1	S_3	S_4	4	3	0
S_3	S_1	S_3	S_4	4	3	4
S_4	S_1	S_6	S_5	1	4	3
S_5	S_1	S_6	S_5	4	4	3
S_6	S_1	S_3	S_4	2	4	4

Составим произвольную входную строку и для нее определим строку внутренних состояний и выходную строку (табл. 2.9).

Таблица 2.9

Состояние	S_0	S_1	S_2	S_4	S_1	S_1	S_1	S_5	S_6	S_1	S_2	S_4	S_1	S_2	S_3	S_4	S_5	S_1	S_5	S_1
Вход	*a*	*b*	*c*	*a*	*a*	*a*	*c*	*b*	*a*	*b*	*c*	*a*	*b*	*b*	*c*	*c*	*a*	*c*	*a*	*b*
Выход	4	4	0	1	3	3	4	4	2	4	0	1	4	3	4	3	4	4	4	4

Предлагаем самостоятельно построить диаграмму состояний графа и проанализировать ее.

Задача 2.4

Описать автомат и построить таблицу и диаграмму состояний, если входной алфавит состоит из символов $\{a, b, c\}$ и автомат печатает 0 – если непосредственно перед этим считал последовательность «*ab*», автомат печатает 1 – если непосредственно перед этим считал последовательность «*ba*», в остальных случаях автомат печатает 2. Автомат работает так только до тех пор, пока не считал последовательно три одинаковых символа *a* или *b* («*aaa*», «*bbb*»), после этого автомат печатает только 2.

Решение.

Составим для удобства таблицу условий работы автомата (табл. 2.10).

Таблица 2.10

Номер условия	Входная последовательность	Реакция автомата (выходной символ)
1	*ab*	0
2	*ba*	1
3	*aaa*	Только 2
4	*bbb*	Только 2
5	Остальные случаи	2

Входной алфавит автомата $A = \{a, b, c\}$, выходной алфавит $Z = \{0, 1, 2\}$.

Для того чтобы обеспечить условия 1 – 2, достаточно запомнить по одному символу («*a*», «*b*» соответственно), так как вторые символы будут считаны с входной ленты. Для обеспечения выполнения условий 3 – 4 необходимо, чтобы автомат имел внутренние состояния, содержащие по три символа («*aaa*», «*bbb*» соответственно). Только в этом случае автомат фактически перестанет реагировать на входную строку и будет печатать только символ 2.

Таким образом, автомат имеет следующие внутренние состояния:

S_0 – начальное состояние (память пуста),

S_1 – «*a*»,

S_2 – «*aa*»,

S_3 – «*aaa*»,

S_4 – «*b*»,

S_5 – «*bb*»,

S_6 – «*bbb*».

Построим таблицу состояний автомата (табл. 2.11).

Таблица 2.11

Текущее состояние	Следующее состояние ν			Выходной символ ξ		
	a	*b*	*c*	*a*	*b*	*c*
S_0	S_1	S_4	S_0	2	2	2
S_1	S_2	S_4	S_0	2	0	2
S_2	S_3	S_4	S_0	2	0	2
S_3	S_3	S_3	S_3	2	2	2
S_4	S_1	S_5	S_0	1	2	2
S_5	S_1	S_6	S_0	1	2	2
S_6	S_6	S_6	S_6	2	2	2

Для проверки описания автомата построим произвольную входную последовательность и определим соответствующие ей последовательности внутренних состояний и выходных символов (табл. 2.12).

Таблица 2.12

Состояние	S_0	S_1	S_4	S_0	S_0	S_4	S_1	S_2	S_3	S_3
Вход	*a*	*b*	*c*	*c*	*b*	*a*	*a*	*a*	*b*	*c*
Выход	2	0	2	2	2	1	2	2	2	2

Построим диаграмму состояний автомата (рис. 2.5).

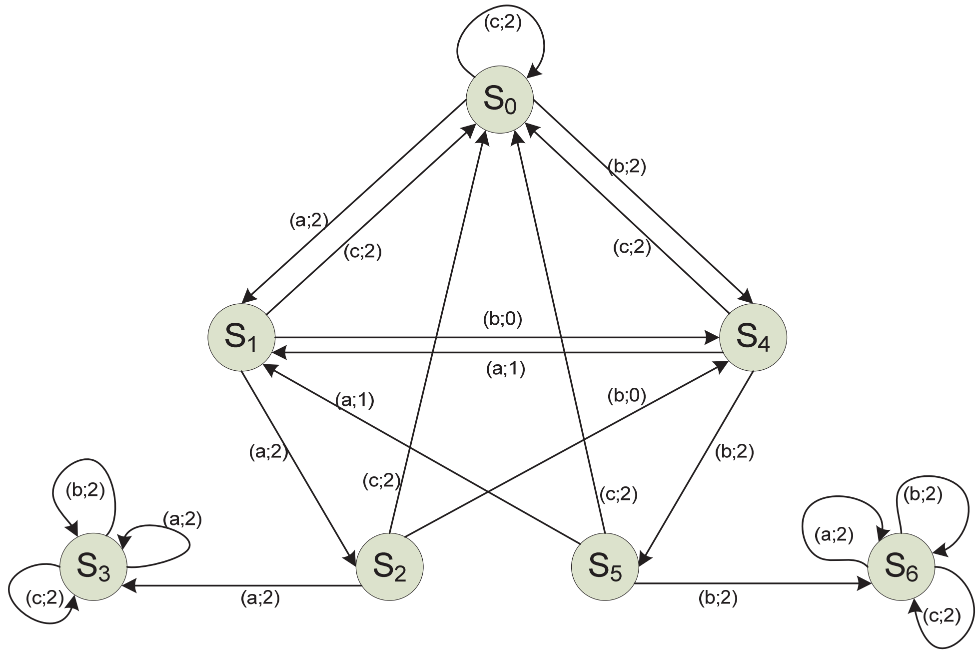


Рис. 2.5

Проанализируем диаграмму состояний автомата:

- автомат может побывать только в одном из состояний: либо S_3, либо S_6;
- состояния S_3 и S_6 являются «зацикленными», т.е. автомат, попав в них, не может выйти из этих состояний и фактически перестает реагировать на входную строку и печатает только символ «2»;
- в начальном состоянии S_0 автомат может находиться неоднократно.

Задача 2.5

Описать автомат, входной алфавит которого содержит два символа 0 и 1. Автомат печатает символ «*», если: не считал ни одной единицы или считал 0. Если автомат считал 1, то он печатает Н – если число считанных единиц нечетное, печатает Ч – если число единиц четное количество.

Решение.

Входной алфавит автомата $A = \{0, 1\}$, выходной алфавит $Z = \{*, Н, Ч\}$.

Для обеспечения условий достаточно ввести всего три внутренних состояния:

S_0 – начальное состояние (память пуста или автомат не считал ни одной единицы);

S_1 – автомат считал нечетное количество единиц;

S_2 – автомат считал четное количество единиц.

Построим таблицу состояний (табл. 2.13).

Таблица 2.13

Текущее состояние	Следующее состояние ν		Выходной символ ξ	
	0	1	0	1
S_0	S_0	S_1	*	Н
S_1	S_1	S_2	*	Ч
S_2	S_2	S_1	*	Н

Построим диаграмму состояний автомата (рис. 2.6).

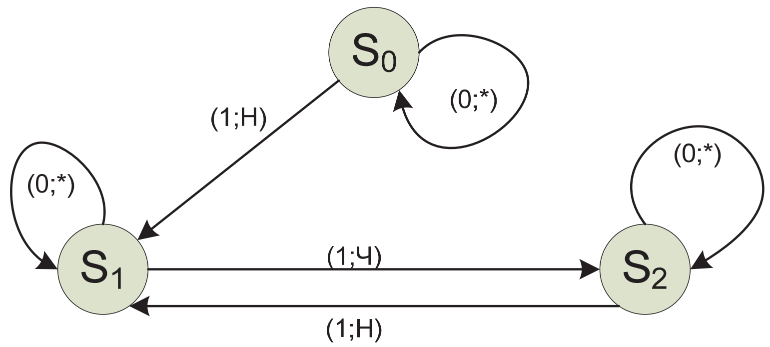


Рис. 2.6

Проанализируем диаграмму состояний автомата:

- состояние S_0 является недостижимым из других состояний, т.е. выйдя из него, автомат больше в него не попадет;
- попав в состояния S_1 и S_2, только при переходе из состояния S_1 в S_2 или наоборот, автомат печатает буквы (Н или Ч), в остальных случаях автомат печатает символ «*».

2.2. Машина Тьюринга

Исторически понятие конечного автомата развилось из близкого понятия, введенного в 1936 г. логиком Аланом Тьюрингом.

Машина Тьюринга является расширением конечного автомата и, согласно тезису Чёрча-Тьюринга, способна имитировать все другие исполнители (с помощью задания правил перехода), каким-либо образом реализующие процесс пошагового вычисления, в котором каждый шаг вычисления достаточно элементарен.

Тьюринг рассматривал гипотетическую «машину», имеющую конечное множество S внутренних состояний и одну бесконечно длинную ленту, разделенную на ячейки, которую машина могла передвигать на одну ячейку вправо или влево за такт. В каждой ячейке машина может записывать символ из конечного алфавита A. Первоначально лента должна быть пустой, кроме конечного числа ячеек, заполненных заранее. (Эти заранее заполненные ячейки можно представлять себе как программу запуска машины.)

В состав машины Тьюринга входит бесконечная в обе стороны *лента* (возможны машины Тьюринга, которые имеют несколько бесконечных лент), разделённая на ячейки, и *управляющее устройство*, способное находиться в одном из *множества внутренних состояний*. Число возможных состояний управляющего устройства конечно и точно задано. Управляющее устройство может перемещаться влево и вправо по ленте, читать и записывать в ячейки ленты символы некоторого конечного алфавита. Выделяется особый *пустой* символ, заполняющий все клетки ленты, кроме тех из них (конечного числа), на которых записаны входные данные. Управляющее устройство работает согласно *правилам перехода*, которые представляют алгоритм, *реализуемый* данной машиной Тьюринга. Каждое правило перехода предписывает машине, в зависимости от текущего состояния и наблюдаемого в текущей клетке символа, записать в эту клетку новый символ, перейти в новое состояние и переместиться на одну клетку влево или вправо. Некоторые состояния машины Тьюринга могут быть помечены как *терминальные*, и переход в любое из них означает конец работы, остановку алгоритма.

Основная разница между машиной Тьюринга и конечным автоматом состоит в том, что:

- лента машины Тьюринга бесконечна;
- машина Тьюринга может двигаться вдоль ленты (или смещать ленту) в любом направлении.

Это придает машине бесконечную память, которой можно пользоваться в ходе вычислений. Сверх того, каждую ячейку можно просматривать многократно.

Формальное определение машины Тьюринга существует в нескольких вариантах. Вот один из них.

Определение 2.2. Машиной Тьюринга называется пятерка объектов следующего типа: $[A, S, \nu, \xi, \delta]$.

Здесь $A = \{a_0, a_1, \ldots, a_n\}$ – *конечный алфавит символов, которые могут быть записаны в ячейках и являются одновременно входными и выходными;*

$S = \{s_0, s_1, \ldots, s_r\}$ – *конечное множество внутренних состояний;*

ν: $S \times A \to S$ – *функция перехода (в следующее состояние);*

ξ: $S \times A \to Z$ – *функция выхода.*

δ: $S \times Rd$ *в множество {Л, П, ОСТАНОВ}.*

Машина Тьюринга работает следующим образом. Она начинает работу, находясь в начальном состоянии s_0. После считывания первого символа она переходит в новое внутреннее состояние, определяемое функцией

v, записывает в ячейке символ, являющийся значением функции ξ, перемещает ленту направо (П), налево (Л) или остается на месте и прекращает работу (ОСТАНОВ) в зависимости от значения функции δ.

На рис. 2.7 схематически изображена лента машины Тьюринга и считывающая/записывающая головка.

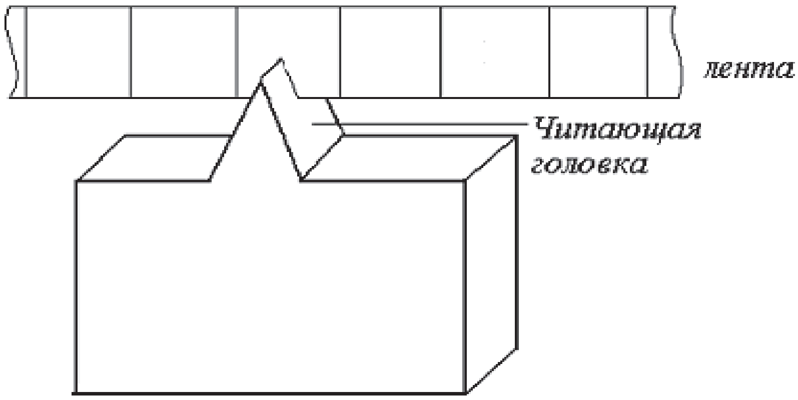


Рис. 2.7

Еще раз укажем, что работа машины состоит в повторении следующего цикла: считывание символа из ячейки, впечатывание нового символа в эту ячейку, выбор которого определяет функция ξ (может оказаться, что это тот же символ), сдвиг ленты налево или направо или остановка. Лента бесконечна в обоих направлениях, однако вначале (и, значит, после любого такта) заполнено лишь конечное число ячеек.

Определение 2.3. Машина Тьюринга называется *детерминированной*, если каждой комбинации состояния и ленточного символа в таблице соответствует не более одного правила. Если существует пара «ленточный символ – состояние», для которой существует две и более команды, такая машина Тьюринга называется *недетерминированной*.

Пример 2.2

Машина Тьюринга, описанная ниже, считывает входную последовательность нулей и единиц, выпечатывает Ч, если число единиц четное, и Н, если нечетное. Строке из нулей и единиц предшествуют и последуют пустые ячейки, обозначаемые #. Символы Ч или Н печатаются в первой пустой ячейке вслед за входной строкой.

Алфавит, таким образом, имеет вид $A = \{\#, 0, 1, Ч, Н\}$. Внутренние состояния $S = \{s_0, s_1, s_2\}$, s_0 – начальное состояние. Машина останавливается по сигналу ОСТАНОВ.

ν: $(S_0, 0) \mapsto s_1$	ξ: $(S_0, 0) \mapsto 0$	δ: $(S_0, 0) \mapsto$ Л
$(S_0, 1) \mapsto s_2$	$(S_0, 1) \mapsto 1$	$(S_0, 1) \mapsto$ Л
$(S_1, 0) \mapsto s_1$	$(S_1, 0) \mapsto 0$	$(S_1, 0) \mapsto$ Л
$(S_1, 1) \mapsto s_2$	$(S_1, 1) \mapsto 1$	$(S_1, 1) \mapsto$ Л
$(S_2, 0) \mapsto s_2$	$(S_2, 0) \mapsto 0$	$(S_2, 0) \mapsto$ Л
$(S_2, 1) \mapsto s_1$	$(S_2, 1) \mapsto 1$	$(S_2, 1) \mapsto$ Л
$(S_0, \#) \mapsto s_0$	$(S_0, \#) \mapsto \#$	$(S_0, \#) \mapsto$ Л
$(S_1, \#) \mapsto s_1$	$(S_1, \#) \mapsto$ Ч	$(S_1, \#) \mapsto$ ОСТАНОВ
$(S_2, \#) \mapsto s_2$	$(S_2, \#) \mapsto$ Н	$(S_2, \#) \mapsto$ ОСТАНОВ

Удобнее задавать функции ν, ξ, δ, пользуясь обозначениями Тьюринга. В этом варианте машина Тьюринга задается конечным множеством пятерок $[s_i, a_i, s_r, z_l, t_n]$. В каждой такой пятерке:

s_i – текущее состояние машины;

a_i – символ, считываемый из ячейки;

s_r – следующее состояние машины;

z_l – символ, печатаемый в ячейке;

t_n – одна из команд П, Л, ОСТАНОВ.

В этих обозначениях описанная выше машина задается так:

S_0	#	S_0	#	Л
S_0	0	S_1	0	Л
S_0	1	S_2	1	Л
S_1	0	S_1	0	Л
S_1	1	S_2	1	Л
S_1	0	S_2	0	Л
S_2	1	S_1	1	Л
S_2	#	S_1	Ч	ОСТАНОВ
S_2	#	S_2	Н	ОСТАНОВ

3. РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ (Часть 2)

3.1. Правила выполнения и оформления расчетно-графического задания

При выполнении расчетно-графического задания (РГЗ) следует строго придерживаться указанных ниже правил.

Выбор варианта осуществляется в соответствии с последней цифрой учебного шифра (номера зачетной книжки) студента. Цифра 0 соответствует варианту 10.

Решение задач следует располагать в порядке следования номеров, указанных в задании, сохраняя номера задач и записывая исходные данные. Если несколько задач имеют общую формулировку, то при оформлении решения общие условия заменяют конкретными данными.

Приступая к выполнению РГЗ, необходимо изучить теоретический материал и ознакомиться с практической частью пособия.

Решения задач следует оформлять аккуратно, подробно объясняя ход решения. В конце работы необходимо привести список использованных источников, указать дату выполнения работы и поставить подпись.

Оформляется РГЗ в соответствии с требованиями РД ФГБОУ ВПО «КнАГТУ» 013-2013 «Текстовые студенческие работы. Правила оформления».

После получения проверенной работы необходимо исправить в ней отмеченные рецензентом ошибки и недочеты.

3.2. Задачи расчетно-графического задания

Задание 8 (*задания 1 – 7 приведены в 1-й части пособия*). Связный граф задан графически (рис. 3.1). Необходимо:

- построить матрицы смежности и инцидентности;
- найти радиус, диаметр и центры графа;
- определить цикломатическое число графа
- используя алгоритм «Фронт волны», найти кратчайший путь из вершины v_1 в v_8.

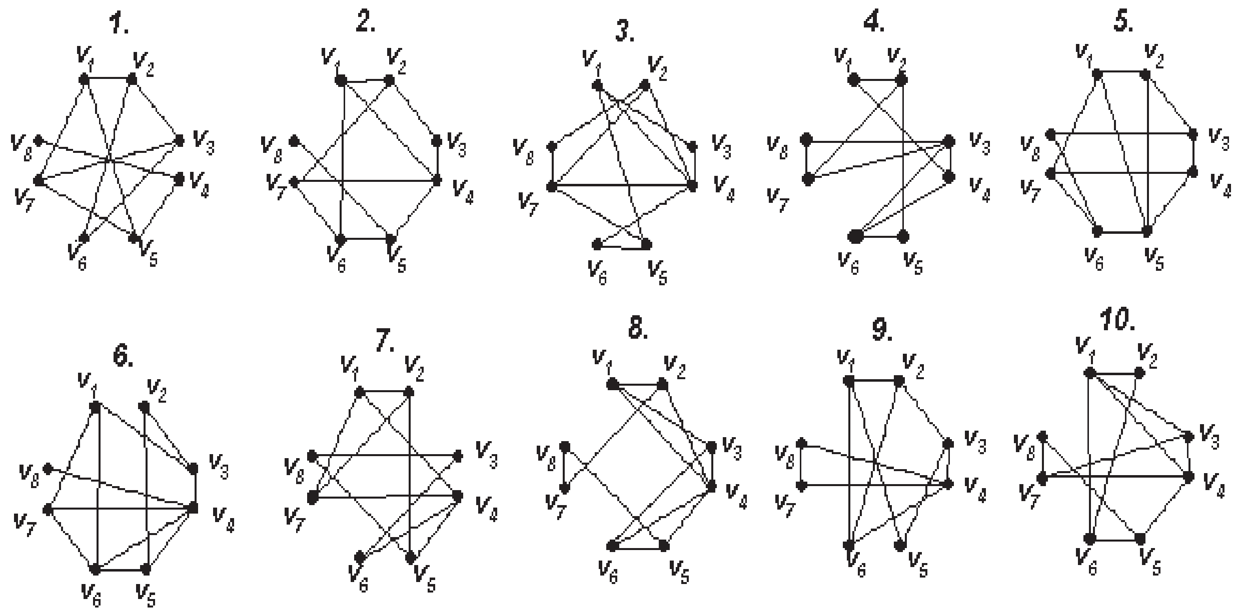


Рис. 3.1

Задание 9. Определить минимальный путь из v_1 в v_8 в нагруженном графе, заданном матрицей длин дуг (табл. 3.1).

Таблица 3.1

1.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	2	∞	∞	3	∞	5	∞
v_2	2	∞	7	∞	∞	9	∞	∞
v_3	∞	7	∞	∞	∞	2	8	4
v_4	∞	∞	∞	∞	∞	8	∞	9
v_5	3	∞	∞	∞	∞	∞	10	∞
v_6	∞	9	2	8	∞	∞	∞	∞
v_7	5	∞	8	∞	10	∞	∞	∞
v_8	∞	∞	4	9	∞	∞	∞	∞

2.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	6	∞	2	∞	7	∞	∞
v_2	6	∞	9	∞	∞	∞	9	∞
v_3	∞	9	∞	8	∞	∞	∞	∞
v_4	2	∞	8	∞	5	∞	7	∞
v_5	∞	∞	∞	5	∞	6	∞	1
v_6	7	∞	∞	∞	6	∞	9	∞
v_7	∞	9	∞	7	∞	9	∞	∞
v_8	∞	∞	∞	∞	1	∞	∞	∞

3.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	∞	6	3	8	∞	∞	∞
v_2	∞	∞	∞	2	∞	∞	1	4
v_3	6	∞	∞	5	∞	∞	∞	∞
v_4	3	2	5	∞	∞	9	7	∞
v_5	8	∞	∞	∞	∞	8	7	∞
v_6	∞	∞	∞	9	8	∞	∞	∞
v_7	∞	1	∞	7	7	∞	∞	10
v_8	∞	4	∞	∞	∞	∞	10	∞

4.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	8	∞	2	∞	∞	∞	∞
v_2	8	∞	∞	∞	9	∞	6	∞
v_3	∞	∞	∞	11	∞	3	10	4
v_4	2	∞	11	∞	∞	5	∞	∞
v_5	∞	9	∞	∞	∞	7	∞	∞
v_6	∞	∞	3	5	7	∞	∞	∞
v_7	∞	6	10	∞	∞	∞	∞	2
v_8	∞	∞	4	∞	∞	∞	2	∞

6.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	∞	6	∞	∞	3	8	∞
v_2	∞	∞	10	∞	6	∞	∞	∞
v_3	6	10	∞	9	∞	∞	∞	∞
v_4	∞	∞	9	∞	3	9	7	7
v_5	∞	6	∞	3	∞	4	∞	∞
v_6	3	∞	∞	9	4	∞	8	∞
v_7	8	∞	∞	7	∞	8	∞	∞
v_8	∞	∞	∞	7	∞	∞	∞	∞

7.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	10	∞	4	∞	∞	∞	∞
v_2	10	∞	∞	∞	11	∞	8	∞
v_3	∞	∞	∞	13	∞	5	12	6
v_4	4	∞	13	∞	∞	7	∞	∞
v_5	∞	11	∞	∞	∞	9	∞	∞
v_6	∞	∞	5	7	9	∞	∞	∞
v_7	∞	8	12	∞	∞	∞	∞	4
v_8	∞	∞	6	∞	∞	∞	4	∞

8.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	2	6	7	∞	∞	∞	∞
v_2	2	∞	∞	5	∞	∞	3	∞
v_3	6	∞	∞	6	∞	6	∞	∞
v_4	7	5	6	∞	8	5	∞	∞
v_5	∞	∞	∞	8	∞	9	∞	9
v_6	∞	∞	6	5	9	∞	∞	∞
v_7	∞	3	∞	∞	∞	∞	∞	4
v_8	∞	∞	∞	∞	9	∞	4	∞

9.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	6	∞	∞	8	4	∞	∞
v_2	6	∞	9	∞	∞	10	∞	∞
v_3	∞	9	∞	10	7	∞	∞	∞
v_4	∞	∞	10	∞	∞	4	8	3
v_5	8	∞	7	∞	∞	∞	∞	∞
v_6	4	10	∞	4	∞	∞	10	∞
v_7	∞	∞	∞	8	∞	10	∞	3
v_8	∞	∞	∞	3	∞	∞	3	∞

5.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	2	∞	∞	9	∞	8	∞
v_2	2	∞	10	∞	2	∞	∞	∞
v_3	∞	10	∞	8	∞	∞	∞	7
v_4	∞	∞	8	∞	6	∞	5	∞
v_5	9	2	∞	6	∞	3	∞	∞
v_6	∞	∞	∞	∞	3	∞	4	2
v_7	8	∞	∞	5	∞	4	∞	∞
v_8	∞	∞	7	∞	∞	2	∞	∞

10.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	∞	6	7	4	∞	8	∞	∞
v_2	6	∞	∞	∞	∞	∞	∞	∞
v_3	7	∞	∞	6	∞	∞	9	5
v_4	4	∞	6	∞	∞	∞	3	∞
v_5	∞	∞	∞	∞	∞	7	∞	∞
v_6	8	∞	∞	∞	7	∞	∞	∞
v_7	∞	∞	9	3	∞	∞	∞	4
v_8	∞	∞	5	∞	∞	∞	4	∞

Задание 10. Для графа из задания 9 построить минимальное остовное дерево.

Задание 11. Для заданной транспортной сети (рис. 3.2) построить полный поток и проверить, является ли он максимальным. Пропускная способность дуг транспортной сети задана в табл. 3.2.

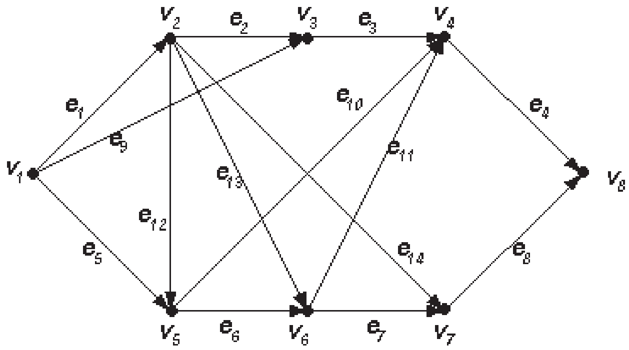


Рис. 3.2

Таблица 3.2

Вари-ант	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}
1	10	3	5	18	10	5	6	7	5	2	6	5	3	3
2	7	8	6	12	9	4	7	7	3	5	2	8	7	2
3	5	8	7	2	3	6	9	4	5	11	9	8	6	4
4	9	2	8	2	5	8	8	7	4	5	6	7	7	10
5	6	5	8	8	7	2	3	4	4	7	10	9	8	9
6	8	5	7	7	8	2	4	10	6	7	5	8	3	4
7	11	10	9	8	7	8	6	5	4	7	2	5	9	11
8	9	4	5	9	5	5	7	8	9	5	2	4	4	3
9	5	9	8	8	7	5	4	8	9	6	5	8	8	7
10	10	3	5	9	6	2	8	5	4	7	9	6	5	7

4. ПРИМЕР ВЫПОЛНЕНИЯ РАСЧЕТНО-ГРАФИЧЕСКОГО ЗАДАНИЯ

Задание 8. Связный граф задан графически (рис. 4.1). Необходимо:

- построить матрицы смежности и инцидентности;
- найти радиус, диаметр и центры графа;
- определить цикломатическое число графа;
- используя алгоритм «Фронт Волны», найти кратчайший путь из вершины v_1 в v_8.

Решение.

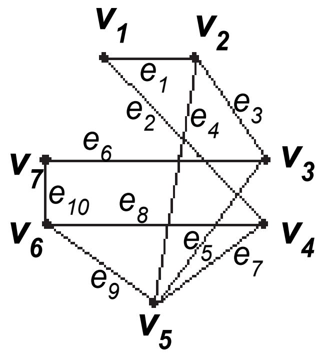


Рис. 4.1

1) Матрица смежности данного графа имеет вид

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Поскольку граф неориентированный, то матрица смежности симметрична относительно главной диагонали.

Поскольку граф неориентированный, то матрица инцидентности состоит только из нулей и единиц. Матрица инцидентности имеет вид:

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

2) Построим матрицу расстояний между вершинами *D(G)* (табл. 4.1):

Радиус графа $R(G) = 2$, диаметр графа $D(G) = 3$, центры графа $\{v_2, v_3, v_4, v_5, v_6\}$.

Таблица 4.1

$D(G)=$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7		$D(v_i)$
v_1	0	1	2	1	2	2	3		3
v_2	1	0	1	2	1	2	2		2
v_3	2	1	0	2	1	2	1		2
v_4	1	2	2	0	1	1	2		2
v_5	2	1	1	1	0	1	2		2
v_6	2	2	2	1	1	0	1		2
v_7	3	2	1	2	2	1	0		3

3) Цикломатическое число графа

$$\nu(G) = m(G) - n(G) + 1 = 10 – 7 + 1 = 4.$$

4) Используем матрицу смежности (табл. 4.2) для нахождения кратчайшего пути из вершины v_1 в v_7.

Таблица 4.2

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	1	0	0	0
v_2	1	0	1	0	1	0	0
v_3	0	1	0	0	1	0	1
v_4	1	0	0	0	1	1	0
v_5	0	1	1	1	0	1	0
v_6	0	0	0	1	1	0	1
v_7	0	0	1	0	0	1	0

Согласно алгоритму «Фронта волны», последовательно определяем:

$FW_1(v_1) = \{v_2, v_4\}$;

$FW_2(v_1) = D(FW_1(v_1))\backslash\{v_1, v_2, v_4\} = \{v_1, v_3, v_5, v_6\}\backslash\{v_1, v_2, v_4\}=\{v_3, v_5, v_6\}$;

$FW_3(v_1) = D(FW_2(v_1))\backslash\{v_1, v_2, v_3, v_4, v_5, v_6\} =$

$= \{v_2, v_5, v_7, v_3, v_4, v_6\}\backslash\{v_1, v_2, v_3, v_4, v_5\} = \{v_7\}$.

Таким образом, $v_7 \in FW_3(v_1)$, а значит, существует путь из v_1 в v_7 длины 3, и этот путь является кратчайшим. Найдем теперь кратчайший путь из v_1 в v_7. Определим множество

$$FW_2(v_1) \cap D^{-1}(v_7) = \{v_3, v_5, v_6\} \cap \{v_3.v_6\} = \{v_3, v_6\}.$$

Выберем любую вершину из найденного множества, например вершину v_3. Определим далее множество

$$FW_1(v_1) \cap D^{-1}(v_3) = \{v_2, v_4\} \cap \{v_2.v_5, v_7\} = \{v_2\}.$$

Тогда $v_1v_2v_3v_7$ – искомый кратчайший путь из v_1 в v_7 (длины 3) в графе G.

Аналогичным образом найдем еще один кратчайший путь. Выберем вершину v_6 из найденного множества $FW_2(v_1) \cap D^{-1}(v_7) = \{v_3, v_6\}$. Тогда

$FW_1(v_1) \cap D^{-1}(v_6) = \{v_2, v_4\} \cap \{v_4 v_5, v_7\} = \{v_4\}$,

Откуда получаем $v_1v_4v_6v_7$ – искомый кратчайший путь из v_1 в v_7 (длины 3) в графе G.

Итак, существует два кратчайших пути из v_1 в v_7 длины 3: $v_1v_2v_3v_7$, $v_1v_4v_6v_7$.

Задание 9. Определить минимальный путь из v_1 в v_7 в нагруженном графе, заданном матрицей длин дуг (табл. 4.3).

Таблица 4.3

	v_1	v_2	v_3	v_4	v_5	v_6	v_7		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$	$\lambda_i^{(6)}$
v_1	∞	∞	5	4	2	2	9		0	0	0	0	0	0	0
v_2	∞	∞	1	1	∞	1	1		∞	∞	6	5	5	5	5
v_3	2	∞	∞	1	1	∞	3		∞	5	4	4	4	4	4
v_4	∞	2	1	∞	1	∞	∞		∞	4	3	3	3	3	3
v_5	∞	∞	2	2	∞	1	6		∞	2	2	2	2	2	2
v_6	1	5	∞	1	1	∞	∞		∞	2	2	2	2	2	2
v_7	2	∞	1	∞	1	2	∞		∞	9	8	8	6	6	6

Решение.

Орграф D задан матрицей $C(D)$ длин дуг нагруженного графа (табл. 3.5). Справа от матрицы $C(D)$ припишем семь столбцов для вычисления λ-матрицы, используя соотношение

$$\lambda_i^{(k+1)} = \min_{1 \le j \le n} \{\lambda_j^{(k)} + c_{ji}\}.$$

Величина $\lambda_7^{(6)} = 6$ выражает длину минимального пути из v_1 в v_7 в нагруженном орграфе D. Найдем минимальное число $k_1 \ge 1$, при котором выполняется равенство $\lambda_7^{(k_1)} = \lambda_7^{(6)}$. Получаем, что $k_1 = 4$. Таким образом, минимальное число дуг в пути среди всех минимальных путей из v_1вv_7 в нагруженном графе D равняется 4. Определим теперь последовательность номеров i_1, i_2, i_3, i_4, i_5, где $i_1 = 7$.

Получаем, что в качестве такой последовательности надо взять номера 7, 6, 4, 2, 1, так как

$$\lambda_7^{(4)} = 6 = \lambda_2^{(3)} + c_{2,7}; i = 2;$$

$$\lambda_2^{(3)} = 5 = \lambda_4^{(2)} + c_{4,2}; i = 4;$$

$$\lambda_4^{(2)} - = 3 = \lambda_6^{(1)} + c_{6,4}; i = 6.$$

Тогда $v_1v_6v_4v_2v_7$ – искомый минимальный путь из v_1 в v_7 в нагруженном орграфе D, равный 6. Причем он содержит минимальное число дуг среди всех возможных минимальных путей из v_1 в v_7.

Задание 10. Для данного графа (рис. 4.2) построить минимальное остовное дерево.

Решение.

На рис. 4.3 приведена последовательность графов, получаемых в результате выполнения алгоритма построения МОД нагруженного графа.

Причем, МОД(G) = 19.

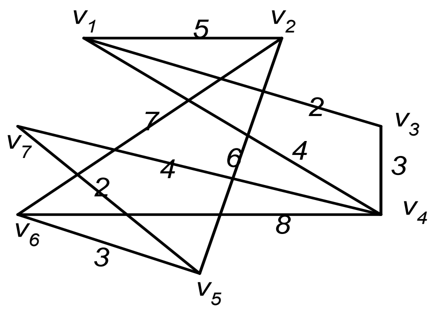


Рис. 4.2

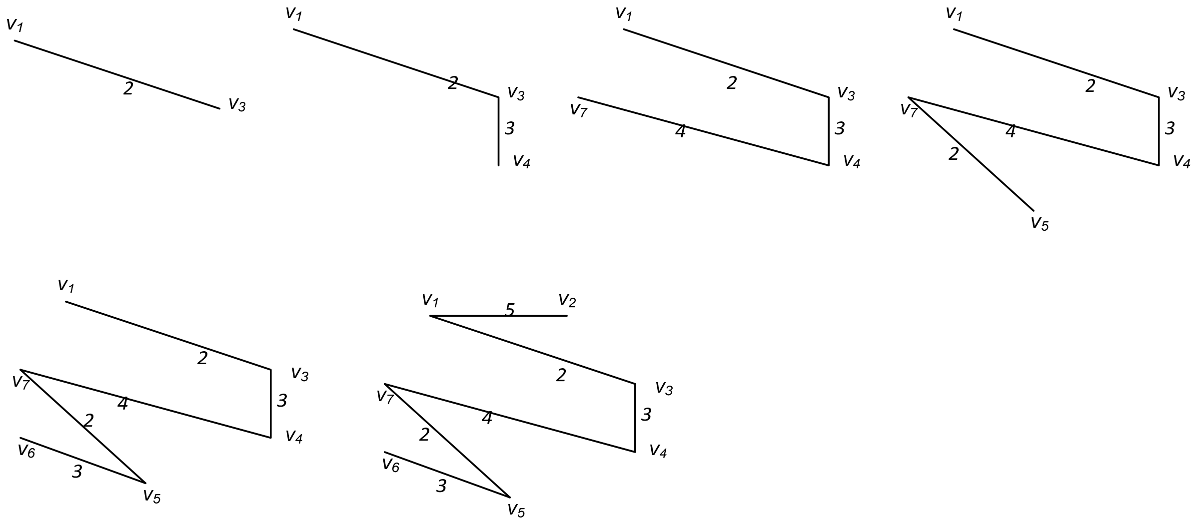


Рис. 4.3

Задание 11. Для заданной транспортной сети (рис. 4.4) построить полный поток и проверить, является ли он максимальным.

Решение.

Начинаем с нулевого потока. Пошагово выделяем простые цепи и увеличиваем поток по ним таким образом, чтобы хотя бы одна дуга в каждой из них стала насыщенной. Полученную насыщенную дугу помечаем пунктиром, помня, что по насыщенной дуге больше идти нельзя.

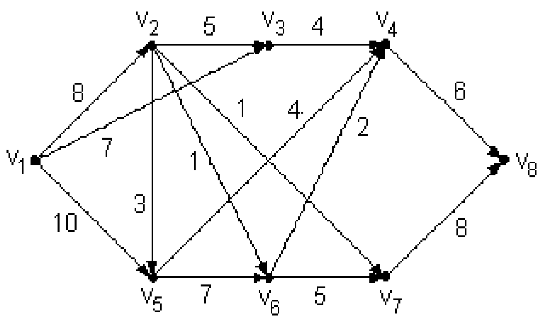


Рис. 4.4

1) $\eta_1 = v_1v_2v_3v_4v_8$, $a_1 = 4$ (рис. 4.5).

2) $\eta_2 = v_1v_5v_6v_7v_8$, a_2 =5 (рис. 4.6).

3) $\eta_3 = v_1v_2v_5v_4v_8$, a_3 = min{4, 3, 4, 2} = 2 (рис. 4.7).

4) $\eta_4 = v_1v_2v_7v_8$, a_1 = min{2, 1, 3} = 1 (рис. 4.8).

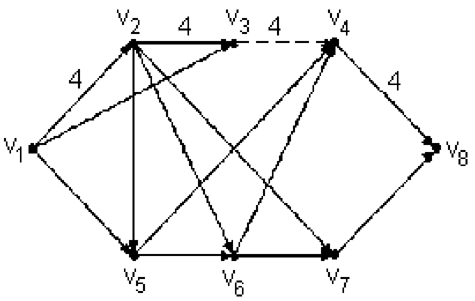


Рис. 4.5

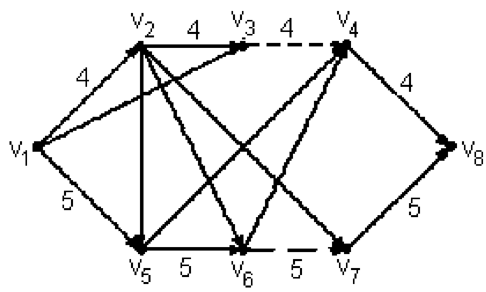


Рис. 4.6

Заметим, что в полученной транспортной сети не существует пути из источника в сток, по которому возможно пройти. Следовательно, построенный поток в транспортной сети является полным и φ = 4 + 5 + 2 + 1 = 12.

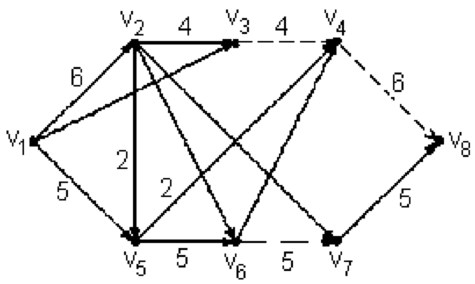


Рис. 4.7

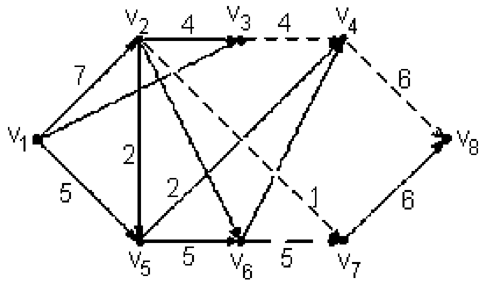


Рис. 4.8

Выясним, является ли построенный полный поток, максимальным. Используем для этого орграф приращений и алгоритм Форда-Беллмана.

Строим матрицу длин дуг орграфа приращений $C(D)$ и λ-матрицу (табл. 4.4).

Таблица 4.4

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8		$\lambda_i^{(0)}$	$\lambda_i^{(1)}$	$\lambda_i^{(2)}$	$\lambda_i^{(3)}$	$\lambda_i^{(4)}$	$\lambda_i^{(5)}$	$\lambda_i^{(6)}$	$\lambda_i^{(7)}$
v_1	∞	0	0	∞	0	∞	∞	∞		0	0	0	0	0	0	0	0
v_2	0	∞	0	∞	0	0	∞	∞		∞	0	0	0	0	0	0	0
v_3	∞	0	∞	∞	∞	∞	∞	∞		∞	0	0	0	0	0	0	0
v_4	∞	∞	0	∞	0	∞	∞	∞		∞	∞	0	0	0	0	0	0
v_5	0	0	∞	0	∞	0	∞	∞		∞	0	0	0	0	0	0	0
v_6	∞	∞	∞	0	0	∞	∞	∞		∞	∞	0	0	0	0	0	0
v_7	∞	0	∞	∞	∞	0	∞	0		∞	∞	∞	∞	∞	∞	∞	∞
v_8	∞	∞	∞	0	∞	∞	0	∞		∞	∞	∞	∞	∞	∞	∞	∞

Так как $\lambda_8^{(7)} = \infty$, то нулевого пути из источника в сток в орграфе не существует. Значит, полный поток φ = 12 является максимальным.

5. КЛЮЧИ К ПРОВЕРОЧНЫМ ТЕСТАМ

5.1. Ключ к проверочному тесту по теме «Теория графов»

Вопрос 1. Коэффициент сложности 6	Ответ: 1
Вопрос 2. Коэффициент сложности 6	Ответ: 2
Вопрос 3. Коэффициент сложности 6	Ответ: 1
Вопрос 4. Коэффициент сложности 6	Ответ: 0
Вопрос 5. Коэффициент сложности 6	Ответ: 2
Вопрос 6. Коэффициент сложности 6	Ответ: 1
Вопрос 7. Коэффициент сложности 6	Ответ: 2
Вопрос 8. Коэффициент сложности 6	Ответ: 3
Вопрос 9. Коэффициент сложности 6	Ответ: 3
Вопрос 10. Коэффициент сложности 6	Ответ: 1
Вопрос 11. Коэффициент сложности 9	Ответ: б) любой простой полный граф с нечетным количеством вершин; в) любой циклический граф
Вопрос 12. Коэффициент сложности 1	Ответ: б) $\upsilon = m - n + k$
Вопрос 13. Коэффициент сложности 9	Ответ: г) $FW_3(V_1) = \{V_8\}$
Вопрос 14. Коэффициент сложности 5	Ответ: а) связный, ацикличный граф; г) граф, любые две различные вершины которого можно соединить единственной простой цепью
Вопрос 15. Коэффициент сложности 1	Ответ: а) сток

Вопрос 16. Коэффициент сложности 6	Ответ: да
Вопрос 17. Коэффициент сложности 1	Ответ: в) V_1V_2, V_4V_6, V_5V_6
Вопрос 18. Коэффициент сложности 1	Ответ: в) $FW_1(V_1) = \{V_2, V_4\}$
Вопрос 19. Коэффициент сложности 5	Ответ: б) источник
Вопрос 20. Коэффициент сложности 5	Ответ: а) максимальный поток обязательно является полным
Вопрос 21. Коэффициент сложности 1	Ответ: в) вершина является концом или началом ребра
Вопрос 22. Коэффициент сложности 1	Ответ: б) для поиска кратчайшего пути из вершины V_1 в вершину V_n .
Вопрос 23. Коэффициент сложности 5	Ответ: г) выражает величину минимального пути из v_1 в v_{i_1} в нагруженном орграфе
Вопрос 24. Коэффициент сложности 5	Ответ: а) величине, равной сумме потоков по всем дугам, заходящим в сток; в) величине, равной сумме потоков по всем дугам, исходящим из источника
Вопрос 25. Коэффициент сложности 1	Ответ: а) 0
Вопрос 26. Коэффициент сложности 1	Ответ: б) каждое ребро ориентированно
Вопрос 27. Коэффициент сложности 1	Ответ: в) 9
Вопрос 28. Коэффициент сложности 1	Ответ: в) 2
Вопрос 29. Коэффициент сложности 5	Ответ: б) множество вершин, в которые можно попасть из данной за один шаг
Вопрос 30. Коэффициент сложности 6	Ответ: да
Вопрос 31. Коэффициент сложности 6	Ответ: лес

Вопрос 32. Коэффициент сложности 5	Ответ: в) однородный степени 2 с n вершинами
Вопрос 33. Коэффициент сложности 6	Ответ: 4
Вопрос 34. Коэффициент сложности 5	Ответ: д) для ориентированных и неориентированных графов
Вопрос 35. Коэффициент сложности 5	Ответ: г) 3
Вопрос 36. Коэффициент сложности 9	Ответ: в) вполне несвязный граф с n вершинами
Вопрос 37. Коэффициент сложности 1	Ответ: б) замкнутая простая цепь
Вопрос 38. Коэффициент сложности 5	Ответ: б) если сумма весов ребер, составляющих этот путь, является наименьшей по сравнению с другими путями
Вопрос 39. Коэффициент сложности 1	Ответ: а) V_2, V_4, V_5
Вопрос 40. Коэффициент сложности 6	Ответ: простой
Вопрос 41. Коэффициент сложности 5	Ответ: а) такая вершина, что максимальное расстояние между ней и любой другой вершиной является наименьшим из всех возможных
Вопрос 42. Коэффициент сложности 5	Ответ: а) сколько ребер необходимо удалить, чтобы граф стал деревом
Вопрос 43. Коэффициент сложности 5	Ответ: б) любой его подграф, содержащий все вершины графа G и являющийся деревом
Вопрос 44. Коэффициент сложности 1	Ответ: б) поток по ней равен ее пропускной способности
Вопрос 45. Коэффициент сложности 9	Ответ: а) любой простой полный граф; б) любой простой полный граф с нечетным количеством вершин; в) любой циклический граф; г) колесо.

Вопрос 46. Коэффициент сложности 1	Ответ: г) 7
Вопрос 47. Коэффициент сложности 5	Ответ: б) искомого пути не существует
Вопрос 48. Коэффициент сложности 5	Ответ: г) $\lambda_n^{(n-1)} = \infty$
Вопрос 49. Коэффициент сложности 5	Ответ: а) путь, содержащий наименьшее из возможных количество ребер
Вопрос 50. Коэффициент сложности 1	Ответ: д) если каждому ребру соответствует целое положительное число
Вопрос 51. Коэффициент сложности 9	Ответ: б) это минимальная величина пути из первой в i-ю вершину и содержит не более k ребер
Вопрос 52. Коэффициент сложности 5	Ответ: а) если любой путь из источника в сток содержит, по крайней мере, одну насыщенную дугу
Вопрос 53. Коэффициент сложности 5	Ответ: б) дерево с минимальной суммой весов содержащихся в нем ребер
Вопрос 54. Коэффициент сложности 8	Ответ: 1 – а, 2 – б, 3 – в
Вопрос 55. Коэффициент сложности 8	Ответ:1 – а, 2 – б, 3 – в
Вопрос 56. Коэффициент сложности 8	Ответ:1 – а, 2 – б, 3 – в
Вопрос 57. Коэффициент сложности 8	Ответ:1 – а, 2 – б, 3 – в
Вопрос 58. Коэффициент сложности 2	Ответ: 10
Вопрос 59. Коэффициент сложности 2	Ответ: 6
Вопрос 60. Коэффициент сложности 2	Ответ: 7
Вопрос 61. Коэффициент сложности 2	Ответ: 9
Вопрос 62. Коэффициент сложности 2	Ответ: 7

Вопрос 63. Коэффициент сложности 7	Ответ: 1) v_1v_2, 2) v_2v_3, 3) v_3v_4
Вопрос 64. Коэффициент сложности 7	Ответ: 1) v_1v_2, 2) v_2v_3, 3) v_3v_4
Вопрос 65. Коэффициент сложности 7	Ответ: 1) v_1v_4, 2) v_1v_3, 3) v_2v_3
Вопрос 66. Коэффициент сложности 7	Ответ: 1) v_1v_2, 2) v_1v_3, 3) v_1v_4
Вопрос 67. Коэффициент сложности 7	Ответ: 1) v_3v_4, 2) v_1v_3, 3) v_1v_2
Вопрос 68. Коэффициент сложности 4	Ответ:1 – а, 2 – б, 3 – в
Вопрос 69. Коэффициент сложности 4	Ответ:1 – а, 2 – б, 3 – в
Вопрос 70. Коэффициент сложности 4	Ответ:1 – а, 2 – б, 3 – в
Вопрос 71. Коэффициент сложности 2	Ответ: насыщенной
Вопрос 72. Коэффициент сложности 10	Ответ: две
Вопрос 73. Коэффициент сложности 10	Ответ: четную
Вопрос 74. Коэффициент сложности 12	Ответ:1 – а, 2 – б, 3 – в
Вопрос 75. Коэффициент сложности 3	Ответ: а), б), в)
Вопрос 76. Коэффициент сложности 11	Ответ: а), б), в), г), д)

5.2. Ключ к тесту по теме «Сетевое планирование и управление»

Вопрос 1. Коэффициент сложности 4	Ответ: 1 – б; 2 – а; 3 – в
Вопрос 2. Коэффициент сложности 4	Ответ: 1 – б; 2 – а; 3 – в

Вопрос 3. Коэффициент сложности 2	Ответ: 1
Вопрос 4. Коэффициент сложности 2	Ответ: 0
Вопрос 5. Коэффициент сложности 2	Ответ: 0
Вопрос 6. Коэффициент сложности 2	Ответ: 2
Вопрос 7. Коэффициент сложности 1	Ответ: г) $V_0 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5$
Вопрос 8. Коэффициент сложности 3	Ответ: а – в – б – г
Вопрос 9. Коэффициент сложности 4	Ответ: 1 – а; 2 – в; 3 – б
Вопрос 10. Коэффициент сложности 4	Ответ:1 – а; 2 – в; 3 – б
Вопрос 11. Коэффициент сложности 2	Ответ: 0
Вопрос 12. Коэффициент сложности 2	Ответ: 3
Вопрос 13. Коэффициент сложности 2	Ответ: 0
Вопрос 14. Коэффициент сложности 2	Ответ: 0
Вопрос 15. Коэффициент сложности 2	Ответ: 1
Вопрос 16. Коэффициент сложности 1	Ответ: в) $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_6$
Вопрос 17. Коэффициент сложности 3	Ответ: а – г – б – в – д
Вопрос 18. Коэффициент сложности 2	Ответ: 1
Вопрос 19. Коэффициент сложности 2	Ответ: 0

Вопрос 20. Коэффициент сложности 2	Ответ: 0
Вопрос 21. Коэффициент сложности 2	Ответ: 1
Вопрос 22. Коэффициент сложности 2	Ответ: 1
Вопрос 23. Коэффициент сложности 1	Ответ: а) $V_0 \to V_2 \to V_3 \to V_6$
Вопрос 24. Коэффициент сложности 3	Ответ: а – в – б – г
Вопрос 25. Коэффициент сложности 12	Ответ:1 – в; 2 – а; 3 – б
Вопрос 26. Коэффициент сложности 12	Ответ:1 – а; 2 – в; 3 – б
Вопрос 27. Коэффициент сложности 10	Ответ: 0
Вопрос 28. Коэффициент сложности 10	Ответ: 5
Вопрос 29. Коэффициент сложности 10	Ответ: 3
Вопрос 30. Коэффициент сложности 10	Ответ: 0
Вопрос 31. Коэффициент сложности 10	Ответ: 0
Вопрос 32. Коэффициент сложности 10	Ответ: 3
Вопрос 33. Коэффициент сложности 10	Ответ: 0
Вопрос 34. Коэффициент сложности 10	Ответ: 0
Вопрос 35. Коэффициент сложности 10	Ответ: 3
Вопрос 36. Коэффициент сложности 5	Ответ: б) $V_0 \to V_1 \to V_4 \to V_5 \to V_7 \to V_8 \to V_{10}$

Вопрос 37. Коэффициент сложности 7	Ответ: а – г – в – д – е – б – ж
Вопрос 38. Коэффициент сложности 8	Ответ:1 – в; 2 – а; 3 – б
Вопрос 39. Коэффициент сложности 6	Ответ: 0
Вопрос 40. Коэффициент сложности 6	Ответ: 3
Вопрос 41. Коэффициент сложности 6	Ответ: 0
Вопрос 42. Коэффициент сложности 6	Ответ: 3
Вопрос 43. Коэффициент сложности 6	Ответ: 0
Вопрос 44. Коэффициент сложности 6	Ответ: 0
Вопрос 45. Коэффициент сложности 6	Ответ: 1
Вопрос 46. Коэффициент сложности 5	Ответ: г) $V_0 \to V_1 \to V_3 \to V_5 \to V_6 \to V_8$
Вопрос 47. Коэффициент сложности 7	Ответ: а – г – в – д – е – б
Вопрос 48. Коэффициент сложности 9	Ответ: а) $V_0 \to V_1 \to V_6 \to V_7 \to V_8 \to V_9 \to V_{10}$
Вопрос 49. Коэффициент сложности 11	Ответ: б – г – а – е – д – в – ж
Вопрос 50. Коэффициент сложности 9	Ответ: б) $V_0 \to V_3 \to V_4 \to V_9 \to V_{10}$
Вопрос 51. Коэффициент сложности 11	Ответ: а – г – б – в – д

ЗАКЛЮЧЕНИЕ

Теория графов – это несколько особенный раздел дискретной математики. Именно в нем можно найти массу задач и способов их решения как для фундаментальных, так и для прикладных исследований.

Порой в детстве, решая достаточно простые головоломки и «задачи-развлекалочки», мы и не подозревали, что это классические задачи теории графов. В этом и есть красота математики – она живая, она – рядом с нами, она пропитывает всю нашу жизнь. Это подтверждается тем, что в обыденной жизни мы постоянно решаем аналогичные задачи, например, при выборе маршрута путешествий во время отпуска.

Рассмотренные в пособии алгоритмы также имеют широкое применение в экономических системах. Они могут быть использованы при проектировании и совершенствовании логистических систем, в том числе помогают решать задачи оптимизации транспортировки, складирования, распределения.

Изучение теории автоматов позволяет сделать еще один серьезный шаг по развитию строгого логического мышления. Понимание стохастичности строится на следующих моментах: последовательности, взаимосвязи и взаимозависимости внешних и внутренних факторов, влиянии всего этого на реакцию системы. Теория автоматов является одним из основополагающих разделов для имитационного моделирования систем.

Рассмотренные в пособии задачи теории графов и теории автоматов достаточно легко переложить на языки программирования, поэтому они могут быть использованы как практический материал на занятиях по информатике.

Необходимо понимать, что в пособии приведены лишь основы теории графов и теории автоматов. Но, автор надеется, что каждый, кто изучил пособие, получил достаточную базу для углубления и расширения своих познаний в области дискретной математики.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акимов, О. Е. Дискретная математика: логика, группы, графы / О. Е. Акимов. – М. : Лаборатория базовых знаний, 2001. – 376 с.

2. Бронштейн, И. Н. Справочник по математике для инженеров и учащихся втузов / И. Н. Бронштейн, К. А. Семендяев. – М. : Наука, 1986. – 545 с.

3. Воротников, С. М. Введение в математическую логику: методические указания и задания к контрольным работам / С. М. Воротников. – Комсомольск-на-Амуре : Комсомольский-на-Амуре гос. техн. ун-т, 1996. – 128 с.

4. Иванов, Б. Н. Дискретная математика: алгоритмы и программы / Б. Н. Иванов. – М. : Лаборатория базовых знаний, 2001. – 282 с.

5. Москинова, Г. И. Дискретная математика: математика для менеджера / Г. И. Москинова – М. : Логос, 2000. – 240 с.

6. Набебин, А. А. Логика и Пролог в дискретной математике / А. А. Набебин. – М. : Изд-во МЭИ, 1996. – 452 с.

7. Нефедов, В. Н. Курс дискретной математики / В. Н. Нефедов, В. А. Осипова. – М. : Изд-во МАИ, 1992. – 264 с.

8. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – 3-е изд. – СПб. : Питер, 2009. – 384 с.

9. Шапиро, С. В. Решение логических и игровых задач / С. В. Шапиро. – М. : Радио и связь, 1984. – 152 с.

10. Судоплатов, С. В. Дискретная математика : учебник для вузов / С. В. Судоплатов, Е. В. Овчинникова. – 2-е изд., перераб. – М. : ИНФРА-М ; Новосибирск : НГТУ, 2007. – 255 с. (Высшее образование).

11. Горбатов, В. А. Фундаментальные основы дискретной математики. Информационная математика / В. А. Горбатов. – М. : Наука. Физматлит, 2000. – 544 с.