ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Высшая математика»

ДИСКРЕТНАЯ МАТЕМАТИКА

Методические указания к практическим занятиям по теме «Теория графов» для студентов специальности 20 01 02 «Приборы и методы контроля качества и диагностика»

Могилев 2010

УДК 519 (075.8)
ББК 22.176
Д 48

Рекомендовано к опубликованию
учебно-методическим управлением
ГУ ВПО «Белорусско-Российский университет»

Одобрено кафедрой «Высшая математика» «28» января 2009 г., протокол № 5

Составитель А. Г. Козлов

Рецензент канд. техн. наук, доц. С. С. Сергеев

Методические указания к практическим занятиям содержат основные сведения по теории графов и приложения, а также задания для самостоятельного выполнения.

Учебное издание

ДИСКРЕТНАЯ МАТЕМАТИКА

Ответственный за выпуск	Л. В. Плетнев
Технический редактор	А. Т. Червинская
Компьютерная верстка	И. А. Алексеюс

Подписано в печать 11.02.2010. Формат 60х84/16. Бумага офсетная. Гарнитура Таймс. Печать трафаретная. Усл.-печ. л. 2.09. Уч.-изд. л. 2.0. Тираж 56 экз. Заказ № 108.

Издатель и полиграфическое исполнение
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет»
ЛИ № 02330/375 от 29.06.2004 г.
212000, г. Могилев, пр. Мира, 43

1 Основные понятия

Многие системы характеризуются различными по форме связями, а также различными видами взаимодействия. В общем случае взаимодействия могут реализоваться как посредством реальных элементов, таких как железнодорожные вагоны, автомобили, так и с помощью некоторых элементов, не имеющих вещественного выражения (например, распространение эпидемии, дружба или наследственность). Система автомобильных дорог, телефонная сеть, торговая система складов и магазинов, электросеть или сеть авиалиний представляют собой примеры систем, в которых осуществляется то или иное взаимодействие элементов, по сети передается тот или иной поток. Часто математической моделью этих сетей может служить граф.

Граф можно рассматривать как набор точек, называемых вершинами, соединенных между собой линиями, называемыми ветвями (рисунок 1). Представление ряда физических систем в виде графов является совершенно естественным, в то время как в других случаях связь между самой системой и ее моделью в виде графа может быть чрезвычайно сложной.

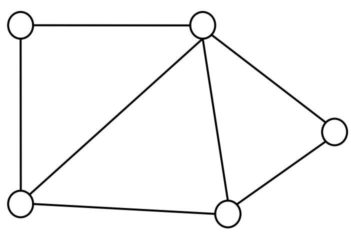


Рисунок 1 – Графическое представление графа

Графом называют двойку $G=(V,E)$, где $V=\{v_1,v_2,...,v_n\}$ – множество элементов, называемых вершинами, $E=\{e_1,e_2,...,e_m\}$ – множество ребер (дуг) графа. Каждое ребро (дуга) из *E* определяется парой вершин v_i и v_j, которые оно соединяет. Такие вершины называют концевыми для ребра (дуги). Ребро графа *G* задается неупорядоченной парой вершин $\{v_i,\ v_j\}$, а дуга – упорядоченной парой вершин $(v_i,\ v_j)$. Дугу графа *G* называют также ориентированным ребром.

Нетрудно видеть, что граф задает бинарное отношение *E*, а множество *V* есть поле данного отношения. Неупорядоченной паре $\{v_i,\ v_j\}$ вершин в отношении *E* соответствуют две упорядоченные пары $(v_i,\ v_j)$ и $(v_j,\ v_i)$.

На рисунке 1 вершины графа обычно изображаются точками (кружочками) на плоскости, а ребра – линиями, соединяющими эти точки (если ребро ориентировано, то на линии указывают направление ориентации). Примеры графов, имеющих ребра или дуги, представлены на рисунке 2.

а) б)

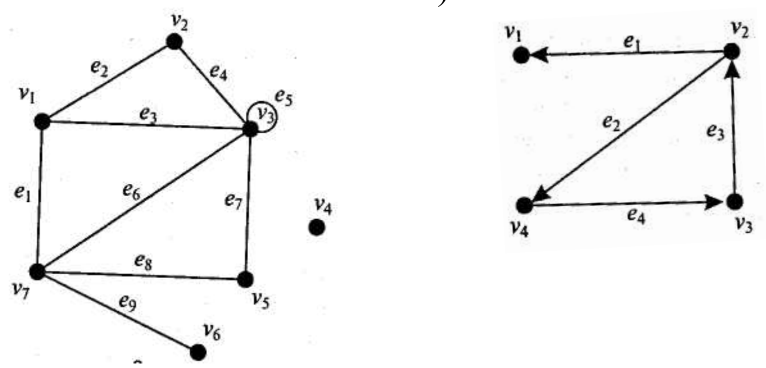


Рисунок 2 – Примеры графов

Граф называют неориентированным, если он не имеет ориентированных ребер (см. рисунок 2, а). Для краткости неориентированный граф называют также неографом.

Иногда каждое ребро такого графа представляют как две дуги, направленные в противоположные стороны.

Граф называют ориентированным, если он состоит только из ориентированных ребер (см. рисунок 2, б), где направление каждого ребра указано стрелкой. Для краткости ориентированный граф называют также орграфом.

Если граф имеет как ориентированные, так и неориентированные ребра, то такой граф называют смешанным.

Ребро называют петлей, если оно начинается и заканчивается в одной и той же вершине. На рисунке 2, а изображена петля, начинающаяся и заканчивающаяся в вершине v_3.

Между вершинами графа G и его ребрами имеет место отношение инцидентности. Говорят, что вершина v_i инцидентна ребру e_k, если v_i – одна из концевых вершин этого ребра. Обратно, всякое ребро инцидентно своим концевым вершинам.

Вершины v_i, v_j $\in V$ графа G называются смежными или соседними, если они инцидентны одному и тому же ребру (соединены хотя бы одним ребром или дугой).

Если хотя бы одну пару вершин графа G соединяют несколько ребер, то такой граф называют мультиграфом.

Вершину графа G называют изолированной, если она не инцидентна ни одной вершине этого графа. Так, вершина v_4 является изолированной в графе (см. рисунок 2, а).

Пусть в графе G имеется дуга $e_k = \left(v_i, v_j\right) \in E$. Тогда говорят, что дуга e_k исходит из вершины v_i и заходит в вершину v_j. Вершину v_i называют также начальной, а вершину v_j – конечной для дуги e_k.

Существуют различные виды графов, которые используются в приложениях.

Граф *G* называют полным, если каждая пара его вершин соединена ребром. Такой граф часто обозначают K_n (рисунок 3, а).

Если граф *G* не имеет ребер, то такой граф называют нуль-графом и обозначают K_0.

Если граф *G* не имеет ни вершин, ни ребер, то его называют пустым, обозначая $K_{\varnothing}$.

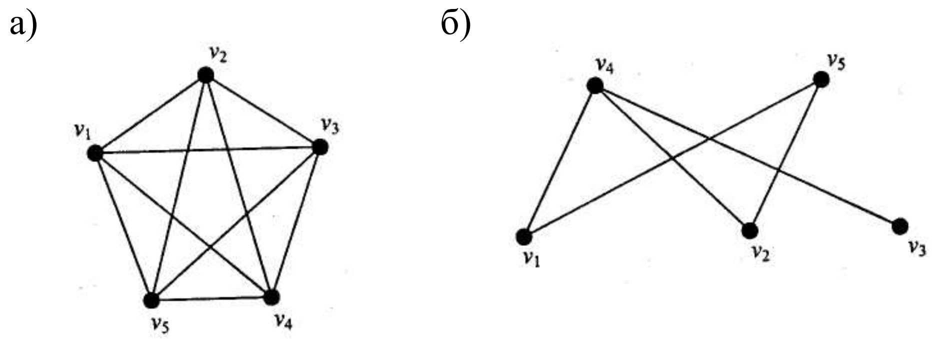


Рисунок 3 – Примеры графов

Граф $G=(V,E)$ называют двудольным, если множество его вершин V можно разбить на два подмножества $V_1, V_2 \subset V$ ($V_1 \cap V_2 = \varnothing$) так, что каждое ребро графа имеет концевые вершины, принадлежащие подмножествам V_1 и V_2. Полный двудольный граф обозначают $K_{m,n}$, где m – число вершин в V_1, а n – число вершин V_2. В этом графе каждые две вершины v_1, v_2 такие, что $v_1 \in V_1$, и $v_2 \in V_2$ соединены ребром.

2 Способы задания графов

Существуют различные способы задания графов, каждый из которых имеет свои преимущества и недостатки.

Рассмотрим наиболее распространенные способы задания графов.

Матрица смежности

Одним из наиболее часто используемых способов задания графа *G* с *n* вершинами является матрица смежности вершин.

Матрица смежности – квадратная матрица, число строк и столбцов которой равно мощности множества $|V| = n$. Элементы матрицы смежности определяются соотношением

$$a_{ij} = \begin{cases} 1, \text{ если } x_i \text{ смежна } x_j; \\ 0, \text{ если } x_i \text{ несмежна } x_j. \end{cases}$$

Для графа, представленного на рисунке 2, а, матрица смежности имеет вид:

$$
\begin{array}{c|ccccccc|c}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & \\
\end{array}
$$

$$
R = \begin{Vmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0
\end{Vmatrix}
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix}
$$

По структуре матрицы смежности ориентированного и неориентированного графов можно сделать следующие выводы:

– каждый ненулевой элемент главной диагонали соответствует петле на графе;

– матрица смежности для неориентированного графа симметрична относительно главной диагонали;

– матрица смежности для ориентированного графа несимметрична относительно главной диагонали;

– если в графе необходимо добавить вершину, несмежную с остальными вершинами, то к матрице смежности следует добавить столбец и строку, элементы которой содержат только «ноль»;

– столбец ориентированного графа, все элементы которого имеют значение «ноль», соответствует вершине-истоку всего графа;

– строка ориентированного графа, все элементы которой имеют значение «ноль», соответствует вершине-стоку всего графа.

Матрица инцидентности

Матрица инцидентности графа G – прямоугольная матрица, число строк которой равно мощности множества отношений $|V| = n$, а число столбцов – мощности множества вершин графа $|E| = m$.

Элементы матрицы инцидентности неориентированного графа определяются по формуле

$$
b_{ij} = \begin{cases} 1, \text{если } x_i \text{ инцедентно } x_j; \\ 0, \text{если } x_i \text{ неинцедентно } x_j. \end{cases}
$$

Следует отметить, что в каждой строке матрицы количество единиц равно двум, а в каждом столбце – равно степени вершины.

Элементы матрицы инцидентности для ориентированного графа определяют по формуле

$$b_{ij}=\begin{cases}1, \text{если дуга } u_j \text{ исходит из вершины } v_i;\\ -1, \text{если дуга } u_j \text{ заходит в вершину } v_i;\\ 0, \text{если дуга } u_j \text{ неинцидентна вершине } v_i.\end{cases}$$

Для графа, представленного на рисунке 2, а, матрица инцидентности имеет вид:

$$Q=\begin{array}{c} \begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{array} \\ \left\|\begin{array}{ccccccccc} 1&1&1&0&0&0&0&0&0\\ 0&1&0&1&0&0&0&0&0\\ 0&0&1&1&1&1&1&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&0&1\\ 1&0&0&0&0&1&0&1&1 \end{array}\right\| \end{array} \begin{array}{c} \\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\\ v_7 \end{array}$$

Список ребер

Список ребер – два массива $g=(g_1,g_2,...,g_m)$ и $h=(h_1,h_2,...,h_m)$, каждый элемент которого есть метка вершины, i-е ребро графа выходит из вершины g_i и входит в вершину h_i.

Список ребер для графа, представленного на рисунке 2, а.

$$g=(1,1,1,2,2,3,3,3,3,5,5,6,7),$$
$$h=(2,3,7,1,3,3,1,5,7,3,7,7,6).$$

Задачи для самостоятельного решения

Задача 1. Дан орграф $G=(V,E)$, где $V=\{v_1,v_2,v_3,v_4,v_5,v_6\}$.

Составить матрицу смежности вершин. Составить матрицу инцидентности. Составить матрицу смежности дуг. Изобразить граф графически.

1 $E=\{(v_2,v_4),(v_2,v_6),(v_2,v_5),(v_3,v_2),(v_4,v_1),(v_5,v_1),(v_5,v_6),(v_5,v_4),$
$(v_6,v_1),(v_6,v_4)\}$

2 $E=\{(v_1,v_2),(v_1,v_3),(v_1,v_6),(v_2,v_3),(v_2,v_5),(v_3,v_4),(v_3,v_6),(v_5,v_3),$
$(v_5,v_4),(v_6,v_4)\}$

3 $E=\{(v_1,v_6),(v_1,v_4),(v_1,v_5),(v_2,v_6),(v_2,v_5),(v_2,v_1),(v_2,v_3),(v_3,v_1),$
$(v_5,v_4),(v_5,v_6)\}$

4 $E=\{(v_2,v_4),(v_3,v_1),(v_4,v_3),(v_5,v_1),(v_5,v_2),(v_5,v_4),(v_5,v_6),(v_6,v_1),$
$(v_6,v_2),(v_6,v_3)\}$

5 $E=\{(v_1,v_3),(v_1,v_4),(v_1,v_5),(v_2,v_1),(v_2,v_5),(v_2,v_6),(v_4,v_3),(v_5,v_4),$
$(v_6,v_1),(v_6,v_5)\}$

Задача 2. Изобразить графы, соответствующие представленным матрицам смежности вершин. Составить матрицы инцидентности этих графов.

$$1\ A=\begin{Vmatrix}0&0&0&0&1&1\\1&0&1&1&0&1\\0&0&0&1&0&0\\0&0&0&0&0&1\\0&0&0&1&0&1\\0&0&0&0&0&0\end{Vmatrix};\ 2\ B=\begin{Vmatrix}1&0&1&1\\0&0&1&1\\0&0&1&0\\0&1&0&0\end{Vmatrix};\ 3\ C=\begin{Vmatrix}0&1&1&0&0\\0&0&1&0&0\\1&0&0&1&1\\0&0&1&0&0\\0&1&1&0&0\end{Vmatrix};$$

$$4\ D=\begin{Vmatrix}1&0&0&1&0\\0&1&1&0&0\\0&1&1&1&1\\1&0&1&1&0\\0&0&1&0&1\end{Vmatrix};\ 5\ E=\begin{Vmatrix}0&0&1&1&1&1\\0&1&0&1&0&1\\1&0&0&0&0&0\\0&0&0&0&0&0\\1&0&1&0&0&0\\0&0&1&1&0&0\end{Vmatrix}$$

Задача 3. Составить матрицу смежности вершин графа (рисунок 4).

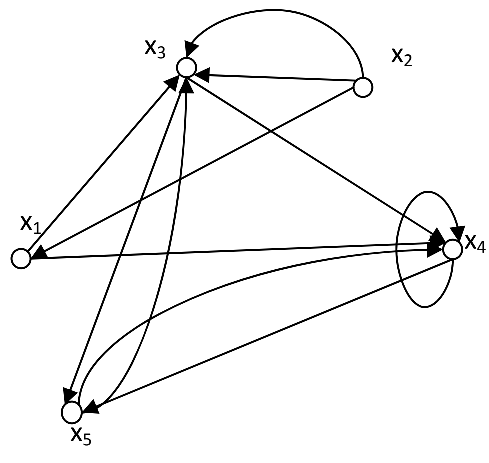


Рисунок 4

Задача 4. Составить матрицу инцидентности графа (рисунок 5).

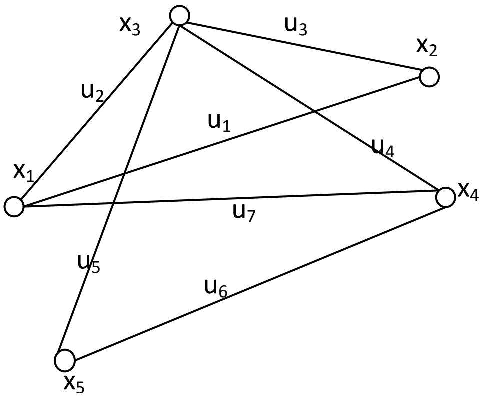
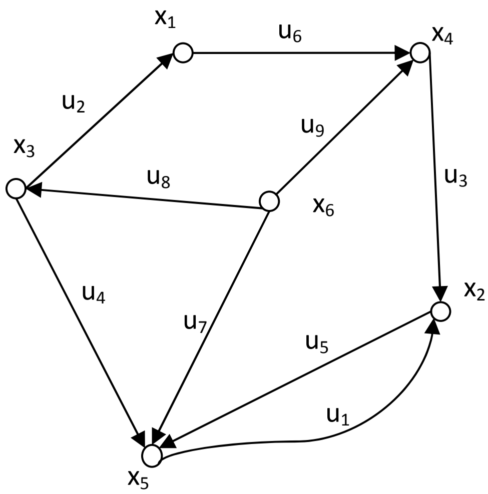


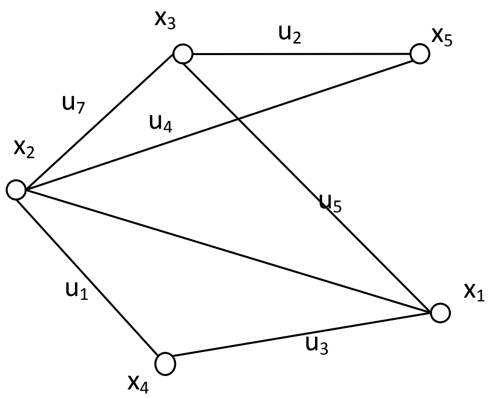
Рисунок 5

Задача 5. Для данных графов составить матрицы смежности вершин, дуг (ребер), инцидентности (рисунок 6).

а)



б)



в)

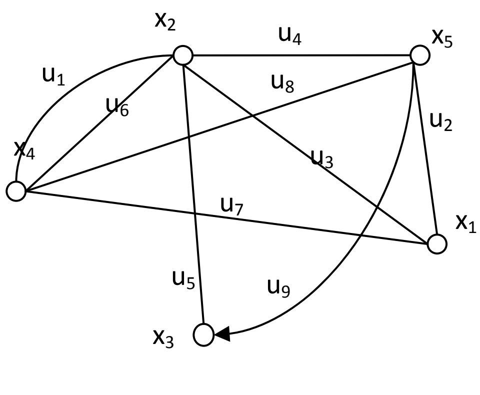


Рисунок 6

Задача 6. Построить наглядное изображение графа по матрице смежности вершин.

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

Задача 7. Построить наглядное изображение графа по матрице смежности дуг.

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

3 Операции на графах

Пусть имеется граф $G = (V, E)$.

Граф $G' = (V', E')$ называется частью графа $G = (V, E)$, если $V' \subset V$ и $E' \subset E$.

Рассмотрим две различные части графа: подграф и суграф.

Граф $G' = (V', E')$ называется подграфом графа $G = (V, E)$, если $V' \subset V$, $E' \subset E$ и подмножество E' графа G' образованно только теми ребрами графа *G,* обе концевые вершины каждого из которых принадлежат множеству V'.

Граф $G' = (V', E')$ называется суграфом графа $G = (V, E)$, если $V' = V$, $E' \subset E$.

Операции над графами позволяют формировать новые графы в результате объединения, пересечения, разности или композиции нескольких графов.

Рассмотрим графы: $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ (рисунок 7).

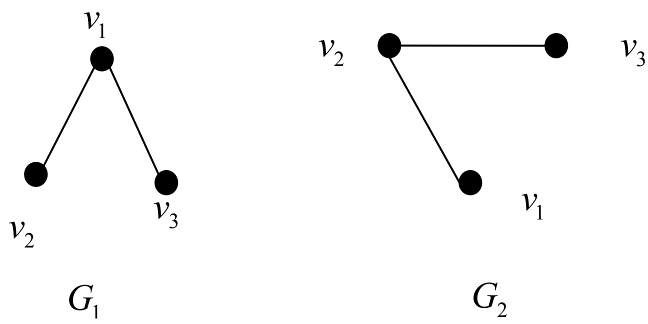


Рисунок 7 – Графы G_1 и G_2

Объединением графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называется граф $G = (V, E) = G_1 \bigcup G_2$, для которого $V = V_1 \bigcup V_2$ и $E = E_1 \bigcup E_2$ (рисунок 8).

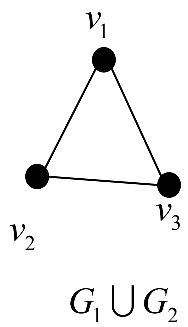


Рисунок 8 – Объединение графов $G_1 \bigcup G_2$

Пересечением графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называется граф $G = (V, E) = G_1 \bigcap G_2$, для которого $V = V_1 \bigcap V_2$ и $E = E_1 \bigcap E_2$ (рисунок 9).

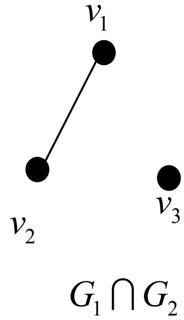
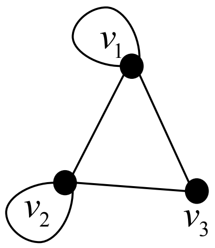


Рисунок 9 – Пересечение графов $G_1 \bigcap G_2$

Композицией графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называется граф $G = (V, E) = G_1 \circ G_2$, в котором имеется ребро $\{v_i, v_j\}$, если имеются ребра

$\{v_i, v_k\} \in E_1$ и $\{v_k, v_j\} \in E_2$ (рисунок 10).



$G_1 \circ G_2$

Рисунок 10 – Композиция графов $G_1 \circ G_2$

Декартовым произведением орграфов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называется орграф $G_1 \times G_2$ с множеством вершин $V_1 \times V_2$, в которых дуга, идущая из вершины (x_1, y_1) в вершину (x_2, y_2) существует тогда и только тогда, когда существует дуга $e_1 = (x_1, x_2)$ и $y_1 = y_2$ или существует дуга $e_2 = (y_1, y_2)$ и $x_1 = x_2$.

Произведением орграфов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называется орграф $G_1 \cdot G_2$ с множеством вершин $V_1 \times V_2$, в которых дуга, идущая из вершины (x_1, y_1) в вершину (x_2, y_2), существует тогда и только тогда, когда существует дуги $e_1 = (x_1, x_2)$ в $G_1(X, E_1)$ и $e_2 = (y_1, y_2)$ в $G_2(Y, E_2)$.

Задачи для самостоятельного решения

Задача 1. Найти декартово произведение следующих графов и произвольный подграф и суграф заданного графа (рисунок 11).

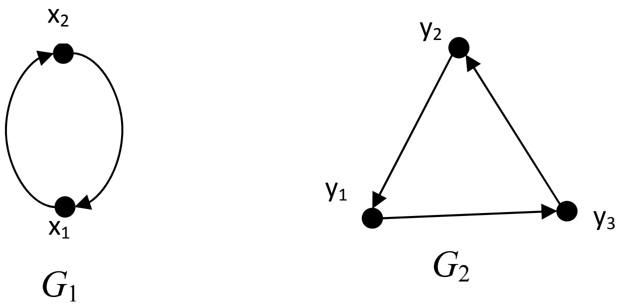


Рисунок 11

Задача 2. Найти произведение графов (рисунок 12).

Задача 3. Найти композицию графов (рисунок 13).

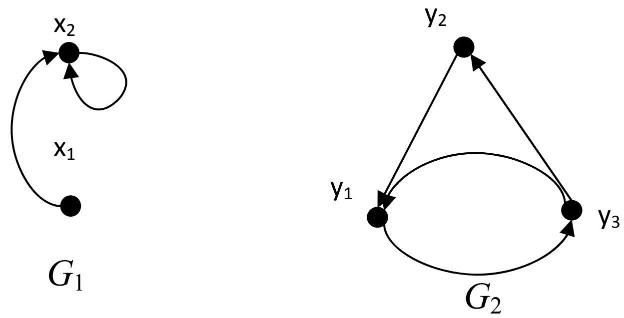


Рисунок 12

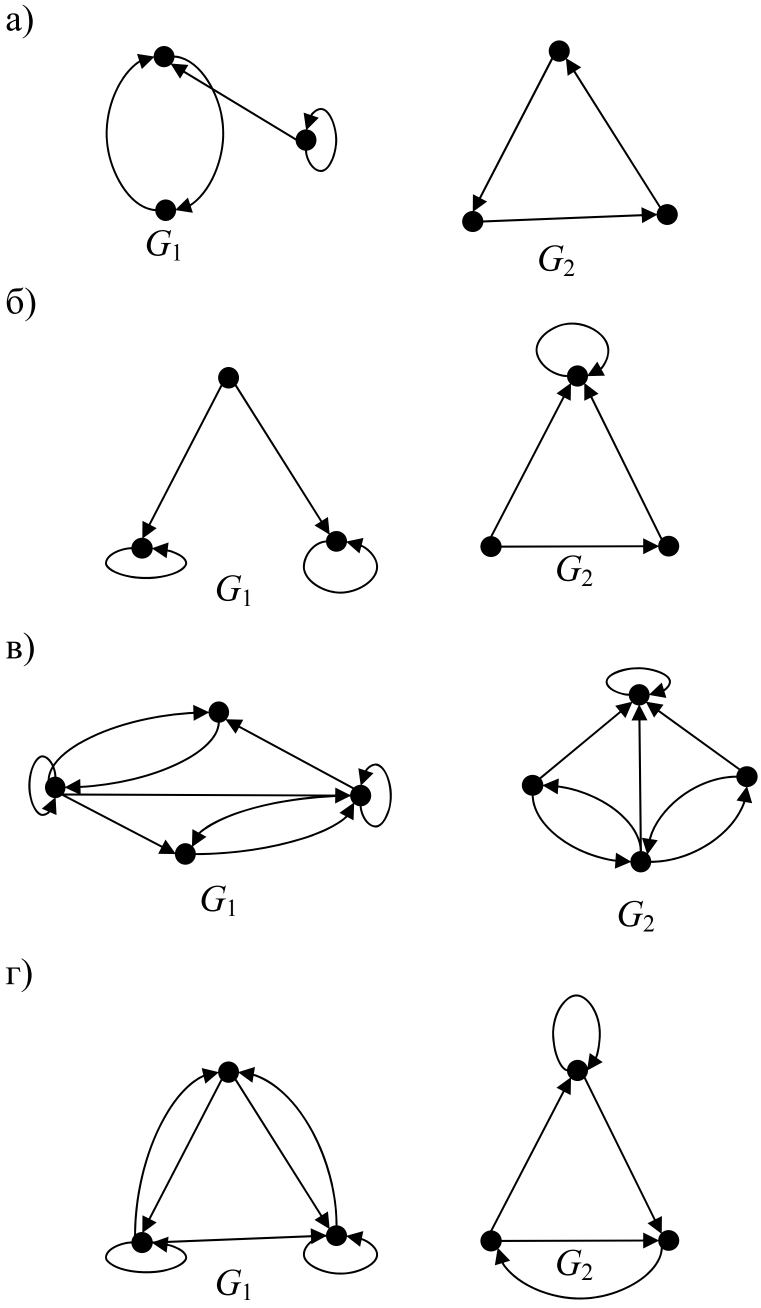


Рисунок 13

4 Числа графов

Любой граф $G=(V,E)$ можно охарактеризовать числами графа (метрическими характеристиками графа). Рассмотрим некоторые из них.

Локальной степенью $\deg(v_i)$ вершины графа v_i называется число ребер, инцидентных данной вершине.

Число дуг орграфа, исходящих из данной вершины, называется полустепенью исхода $\deg^-(v_i)$.

Число руг орграфа, входящих в данную вершину, называется полустепенью захода $\deg^+(v_i)$.

Очевидно, что

$$\deg(v_i)=\deg^-(v_i)+\deg^+(v_i).$$

Цикломатическим числом графа *G* называют число

$$\gamma(G)=m-n+p,$$

где *m* – число ребер;

n – число вершин;

p – число компонент связности графа.

Граф называется связанным, если между любыми его двумя вершинами существует маршрут.

С содержательной точки зрения цикломатическое число определяет число ребер, которые необходимо удалить из графа *G*, чтобы он не имел циклов.

Граф *G* называют *k*-раскрашиваемым, если вершины его можно окрасить, используя *k* различных цветов (красок), таким образом, чтобы концевые вершины каждого ребра имели различные цвета. Наименьшее возможное число *k* называют хроматическим числом графа *G*.

Понятие хроматического числа графа *G* и способы его определения имеют важное практическое значение. Так, большинство задач синтеза конечных автоматов может быть сведено к задаче поиска хроматического числа соответствующего графа.

Любую совокупность вершин $V_1 \subset V_2$ называют независимым множеством вершин графа $G=(V,E)$, если никакие пары вершин из V_1 не смежны (то есть не соединены между собой ребром). Наибольшее по включению независимое множество вершин графа *G* называют максимальным. Максимальное независимое множество вершин называют наибольшим, если оно содержит наибольшее число вершин по сравнению с другими независимыми множествами вершин графа *G*.

Число вершин графа *G* в его наибольшем независимом множестве называют числом независимости или числом внутренней устойчивости графа и обозначают $\alpha_0(G)$.

Любой полный подграф графа *G* называют кликой этого графа. По определению, кликой являются всякая вершина и любое ребро графа *G*.

Число вершин в наибольшей по включению клике называют кликовым числом графа *G*.

Два ребра e_l, $e_k \in E$ графа $G=(V,E)$ называют независимыми, если они не инцидентны одной и той же вершине этого графа. Совокупность попарно независимых ребер графа *G* называют паросочетанием. Паросочетание графа *G* называют наибольшим, если оно имеет наибольшее число ребер по сравнению с другими паросочетаниями этого же графа.

Число ребер в наибольшем паросочетании графа О называют числом паросочетания и обозначают $\alpha_1(G)$.

Подмножество $V_p \subset V$ графа *G* называют вершинным покрытием этого графа, если каждое его ребро имеет хотя бы одну концевую вершину во множестве V_p. Вершинное покрытие называют наименьшим, если оно имеет наименьшее число элементов по сравнению с другими вершинными покрытиями графа *G*.

Число элементов в наименьшем вершинном покрытии называют числом вершинного покрытия и обозначают $\beta_0(G)$.

Подмножество ребер $P \subset E$ графа *G* называют реберным покрытием графа, если всякая вершина этого графа является концевой хотя бы одного ребра из *P*.

Обозначим $d(v_i,v_j)$ расстояние между вершинами v_i и v_j (длина кратчайшего пути между v_i и v_j).

Эксцентриситетом вершины назовем величину $e(v_i)=\max\limits_{v_j\in V} d(v_i,v_j)$.

Максимальный из всех эксцентриситетов назовем диаметром графа *G:*

$$D(G)=\max_{v_i\in V} e(v_i)=\max_{v_i\in V}\max_{v_j\in V} d(v_i,v_j).$$

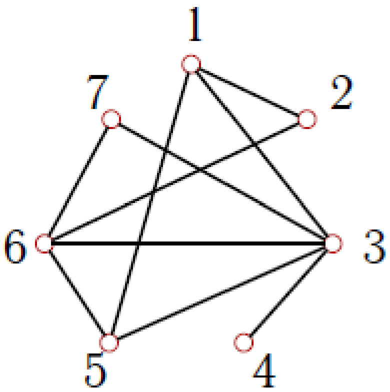
Минимальный из всех эксцентриситетов назовем радиусом графа *G:*

$$R(G)=\min_{v_i\in V} e(v_i)=\min_{v_i\in V}\max_{v_j\in V} d(v_i,v_j).$$

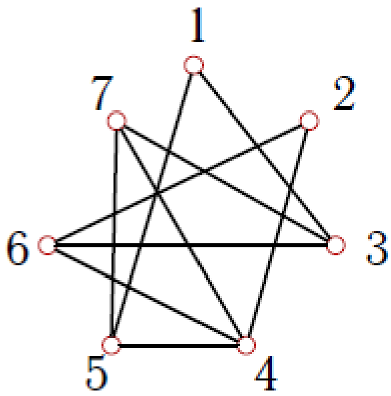
Задачи для самостоятельного решения

Задача 1. Найти радиус и диаметр графа (рисунок 14).

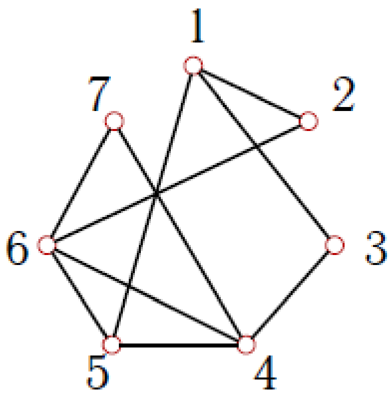
а)



б)



в)



г)

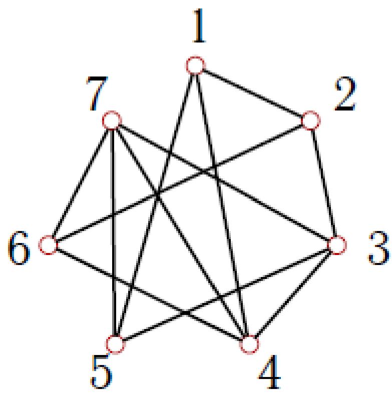
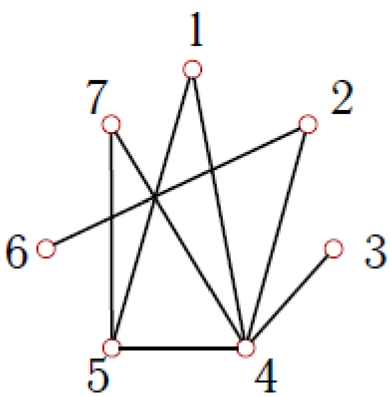


Рисунок 14

Задача 2. Найти раскраски графа (рисунок 15).

а)



б)

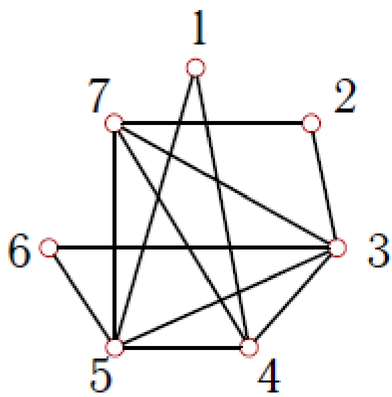
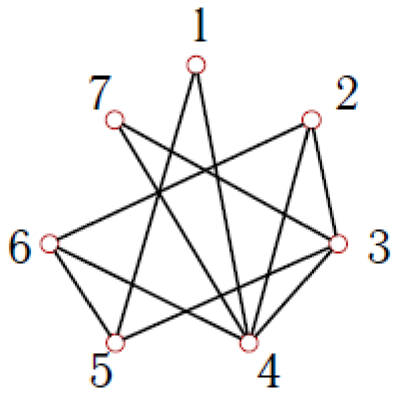
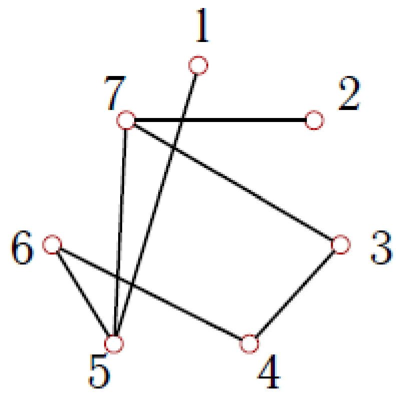


Рисунок 15

в)



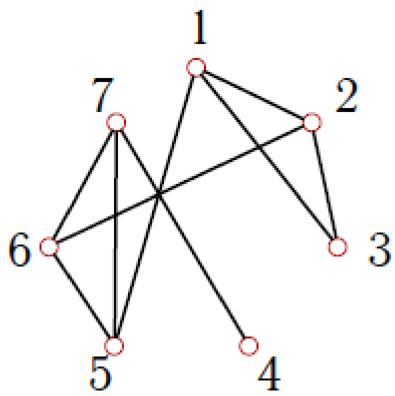
г)



Окончание рисунка 15

Задача 3. Найти степень вершин графа (рисунок 16).

а)



б)

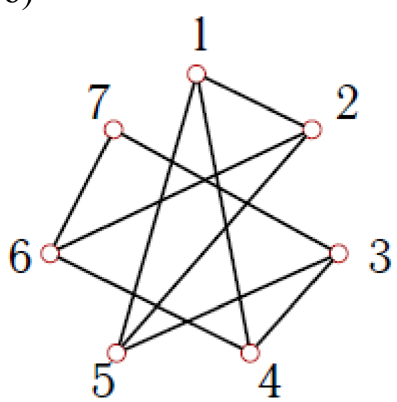


Рисунок 16

5 Отыскание экстремальных путей

Рассмотрим ориентированный граф $G=(V,E)$, дугам которого приписаны веса. Это означает, что каждой дуге $(v_i,v_j)\in E$ поставлено в соответствие некоторое вещественное число $w(v_i,v_j)$, называемое весом данной дуги.

Будем рассматривать нахождение кратчайшего пути между фиксированными вершинами s, $t \in V$. Длину такого кратчайшего пути мы будем обозначать $d(s, t)$ и называть расстоянием от s до t.

Алгоритм Дейкстры

Шаг 1. Положим $d(s)=0$, $y:=s$, $d(x)=\infty$. Помечаем вершину s.

Шаг 2. Для каждой непомеченной вершины x вычисляем новое значение величины $d(x)$ по формуле

$$d(x)=\min\{d(x),d(y)+w(y,x)\}.$$

Шаг 3. Выбираем непомеченную вершину *x*, для которой новое значение $d(x)$ является наименьшим.

Шаг 4. Пометим вершину *x* как рассмотренную и положим $y=x$.

Шаг 5. Если $y=t$, то закончим вычисления, в противном случае, переходим на Шаг 2.

Рассмотрим выполнение алгоритма на примере графа, показанного на рисунке 17. Пусть требуется найти расстояния от первой вершины до всех остальных.

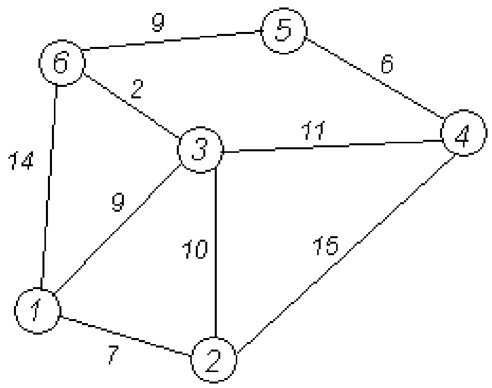


Рисунок 17

Кружками обозначены вершины, линиями – пути между ними (ребра графа). В кружках обозначены номера вершин, над ребрами обозначена их «цена» – длина пути. Рядом с каждой вершиной красным обозначена метка – длина кратчайшего пути в эту вершину из вершины 1 (рисунок 18).

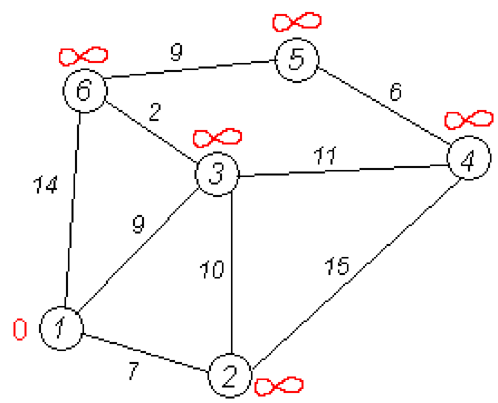


Рисунок 18

Первый шаг. Рассмотрим шаг алгоритма Дейкстры для нашего примера. Минимальную метку имеет вершина 1. Её соседями являются вершины 2, 3 и 6 (рисунок 19).

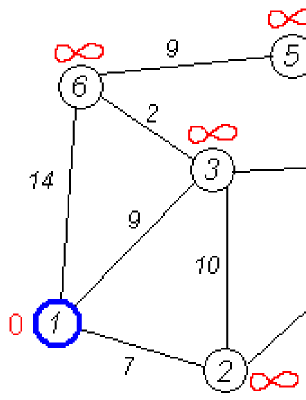


Рисунок 19

Первый по очереди сосед вершины 1 – вершина 2, потому что длина пути до неё минимальна. Длина пути в неё через вершину 1 равна кратчайшему расстоянию до вершины 1 + длина ребра, идущего из 1 в 2, то есть 0 + 7 = 7. Это меньше текущей метки вершины 2, поэтому новая метка второй вершины равна 7 (рисунок 20).

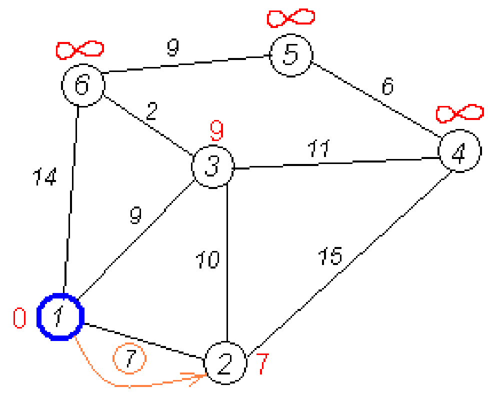


Рисунок 20

Аналогичную операцию проделываем с двумя другими соседями первой вершины – вершинами 3 и 6 (рисунок 21).

Все соседи вершины 1 проверены. Текущее минимальное расстояние до вершины 1 считается окончательным и пересмотру не подлежит. Вычеркнем её из графа, чтобы отметить, что эта вершина посещена (рисунок 22).

а) б)

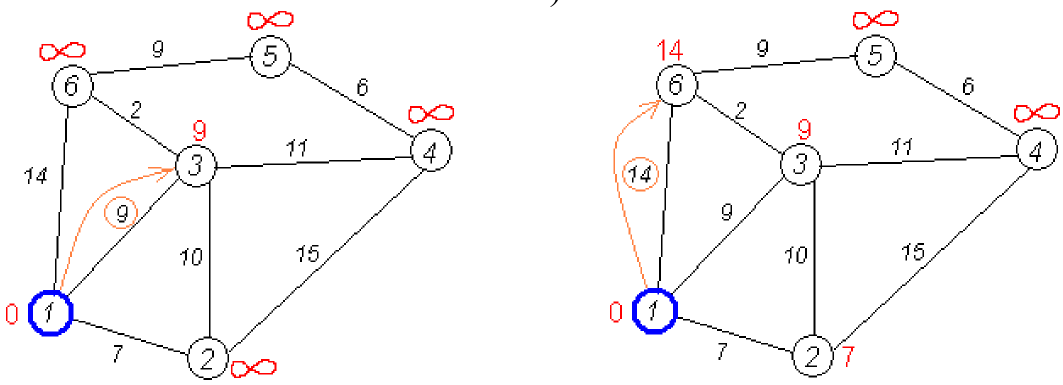


Рисунок 21

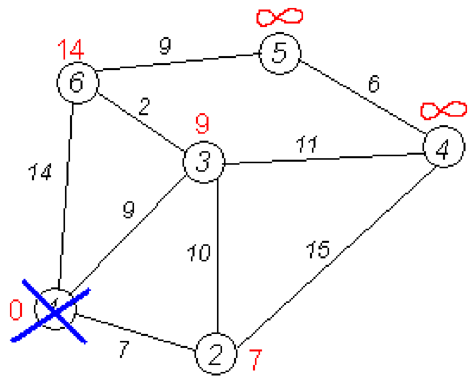


Рисунок 22

Второй шаг. Шаг алгоритма повторяется. Снова находим «ближайшую» из непосещенных вершин. Это вершина 2 с меткой 7 (рисунок 23).

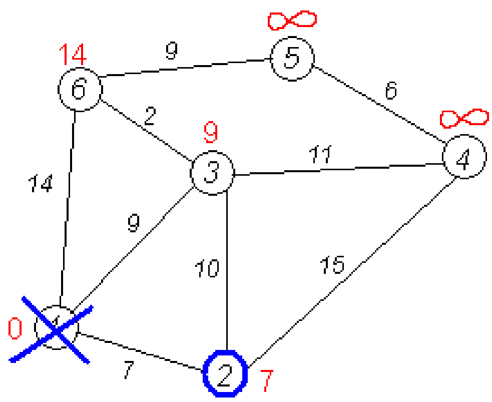


Рисунок 23

Снова пытаемся уменьшить метки соседей выбранной вершины, пытаясь пройти в них через 2-ю. Соседями вершины 2 являются 1, 3, 4.

Первый (по порядку) сосед вершины 2 – вершина 1. Но она уже посещена, поэтому с 1-й вершиной ничего не делаем.

Следующий сосед вершины 2 – вершина 3. Если идти в неё через 2, то длина такого пути будет = 7 + 10 = 17. Но текущая метка третьей вершины равна 9 < 17, поэтому метка не меняется (рисунок 24).

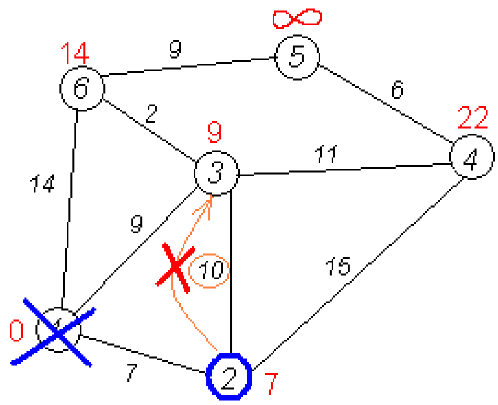


Рисунок 24

Ещё один сосед вершины 2 – вершина 4. Если идти в неё через вторую вершину, то длина такого пути равна кратчайшему расстоянию до 2 плюс расстояние между вершинами 2 и 4. Устанавливаем метку вершины 4 равной 22 (рисунок 25).

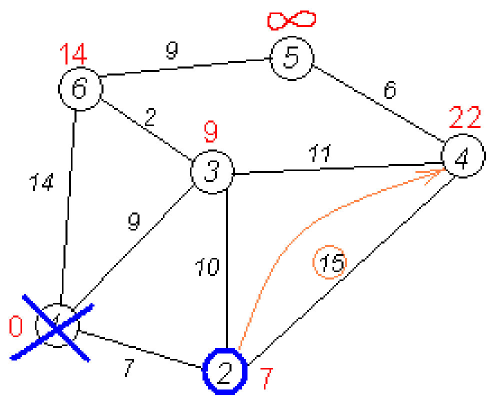


Рисунок 25

Все соседи вершины 2 просмотрены, замораживаем расстояние до неё и помечаем её как посещенную (рисунок 26).

Третий шаг. Повторяем шаг алгоритма, выбрав вершину 3. После её «обработки» получим такие результаты (рисунок 27).

Дальнейшие шаги. Повторяем шаг алгоритма для оставшихся вершин (это будут по порядку 6, 4 и 5) (рисунок 28).

Завершение выполнения алгоритма. Алгоритм заканчивает работу, когда вычеркнуты все вершины. Результат его работы виден на последнем рисунке: кратчайший путь от вершины 1 до 2 составляет 7, до вершины 3 – 9, до вершины 4 – 20, до вершины 5 – 20, до вершины 6 – 11.

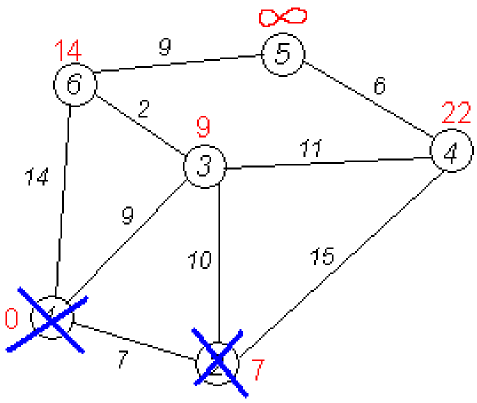


Рисунок 26

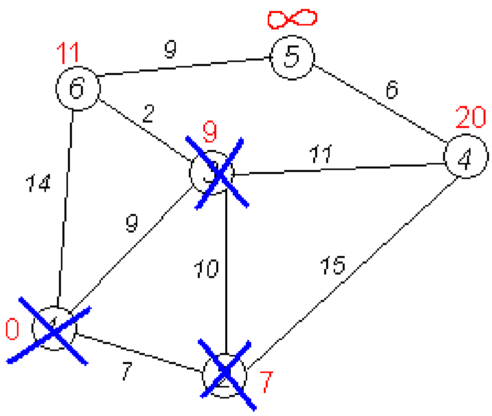


Рисунок 27

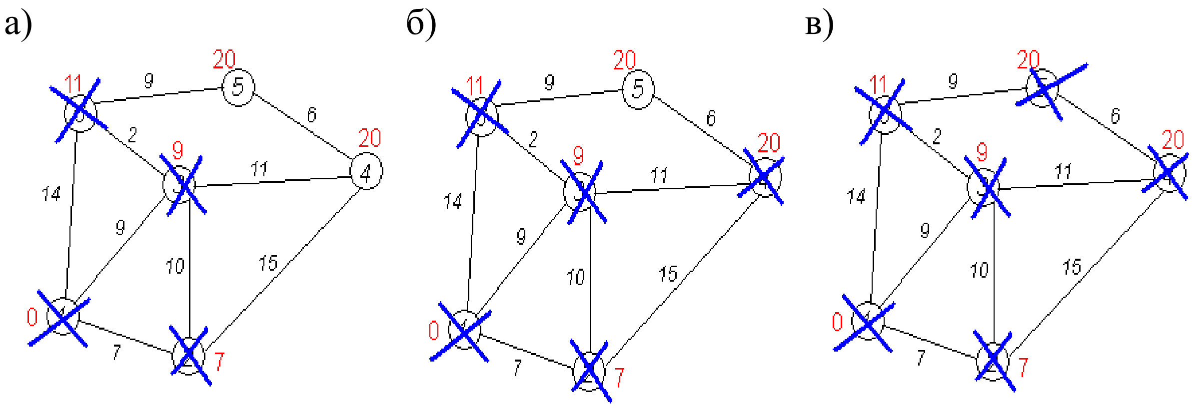


Рисунок 28

Задачи для самостоятельного решения

Задача 1. По заданной матрице весов графа G найти величину минимального пути из вершины x_1 в вершину x_6 по алгоритму Дейкстры, а затем указать сам путь:

$$1\begin{pmatrix} - & 7 & 8 & 13 & \infty & \infty \\ \infty & - & 9 & 7 & 12 & \infty \\ \infty & \infty & - & 6 & 7 & 8 \\ \infty & \infty & \infty & - & 9 & 17 \\ \infty & \infty & \infty & \infty & - & 10 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}; \qquad 2\begin{pmatrix} - & 4 & 5 & 10 & 11 & \infty \\ \infty & - & 11 & 3 & 5 & \infty \\ \infty & \infty & - & 6 & 7 & 18 \\ \infty & \infty & \infty & - & 6 & \infty \\ \infty & \infty & \infty & \infty & - & 8 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix};$$

$$3\begin{pmatrix} - & 8 & \infty & 5 & 10 & \infty \\ \infty & - & 7 & \infty & \infty & \infty \\ \infty & \infty & - & \infty & \infty & 8 \\ \infty & 4 & 6 & - & 5 & 11 \\ \infty & 5 & 5 & \infty & - & 3 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}; \qquad 4\begin{pmatrix} - & \infty & 3 & \infty & \infty & \infty \\ \infty & - & \infty & 4 & 6 & 8 \\ \infty & 8 & - & 5 & 6 & 12 \\ \infty & \infty & \infty & - & 5 & 7 \\ \infty & \infty & \infty & \infty & - & 3 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}.$$

6 Остовы и деревья

Остовом или скелетом связного графа $G=(V,E)$ называется его связный подграф G', содержащий все вершины исходного графа.

Остов, не имеющий циклов, называется остовным деревом графа G.

Пусть ребра связного неориентированного графа G нагружены некоторыми неотрицательными значениями $w(v_i, v_j)$. Минимальным (максимальным) остовным деревом такого нагруженного графа G называется остовное дерево с минимальной (максимальной) суммарной нагрузкой на ребра из всех возможных остовных деревьев.

Для нахождения минимального остовного дерева выберем произвольную вершину и включим ее в строящееся остовное дерево. Затем из всех ребер, смежных с данной вершиной, выберем ребро с минимальной длиной и также включим его в строящееся дерево вместе со второй, инцидентной этому ребру вершиной. Далее на каждом шаге будем искать ребро минимальной длины, ведущее из вершин, уже включенных в строящееся дерево в оставшиеся вершины, и будем поочередно включать найденное ребро в строящееся дерево. Когда в дерево будет включена последняя из вершин графа, работа алгоритма будет закончена.

Пример нахождения остовного дерева (рисунок 29):

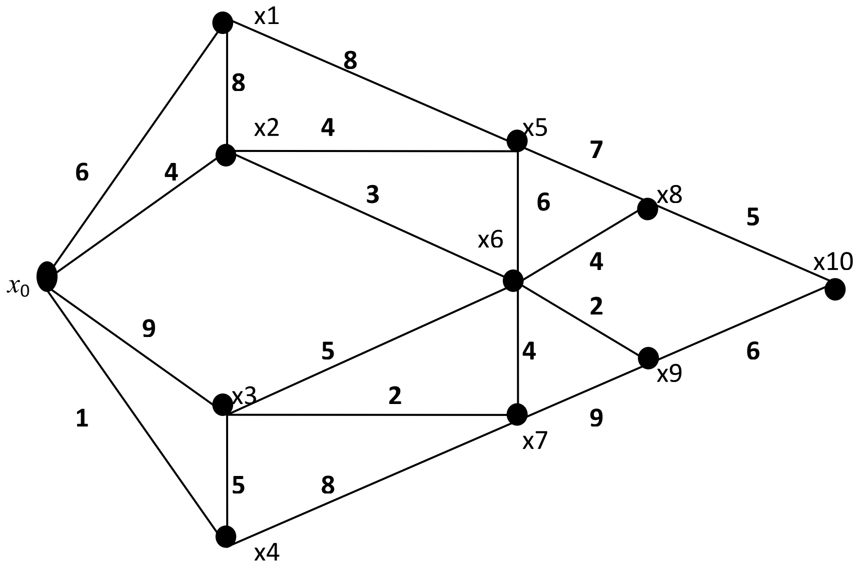


Рисунок 29

Построим максимальное и минимальное остовное дерево для данного нагруженного графа. Для этого составим список рёбер в порядке возрастания и убывания весов графа:

x0x4 = 2, x3x7 = 2, x6x9 = 2,
x2x6 = 3,
x0x2 = 4, x2x5 = 4, x6x7 = 4, x6x8 = 4,
x3x4 = 5, x3x6 = 5, x8x10 = 5,
x0x1 = 6 x5x6 = 6, x9x10 = 6,
x5x8 = 7,
x1x2 = 8, x1x5 = 8, x4x7 = 8,
x0x3 = 9 x7x9 = 9.

Добавим рёбра из списка в граф так, чтобы не образовывалось циклов. Количество рёбер в полученном дереве должно быть (n – 1) = 10.

Минимальное остовное дерево (рисунок 30):

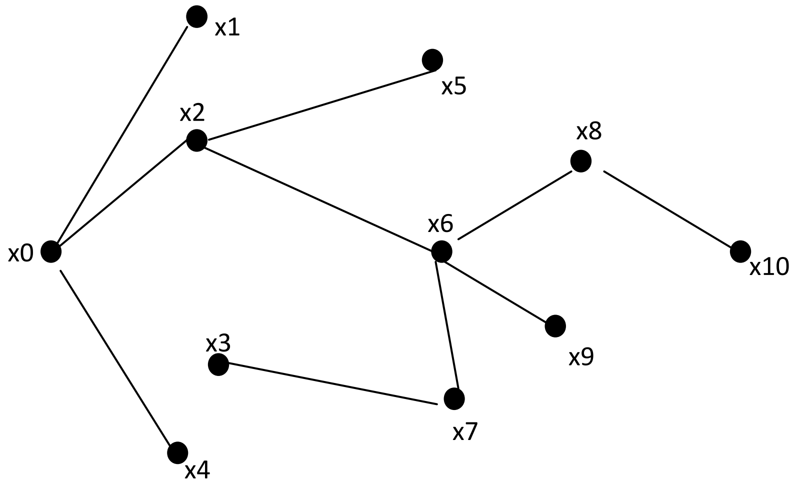


Рисунок 30

Максимальное остовное дерево (рисунок 31):

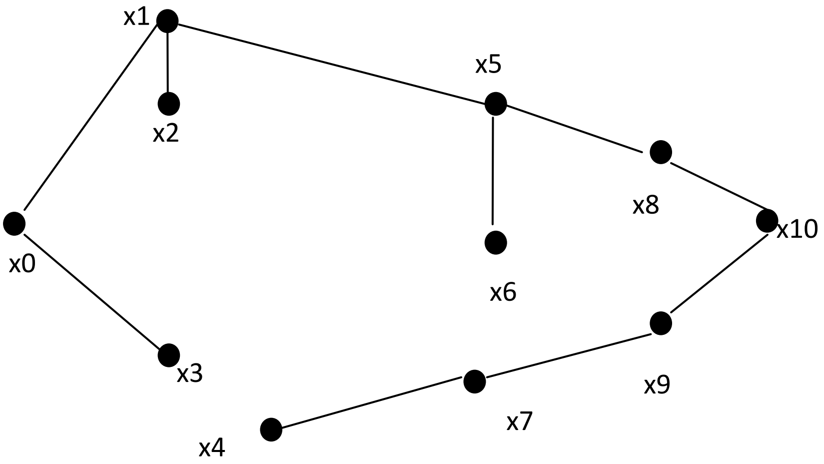


Рисунок 31

Задачи для самостоятельного решения

Задача 1. Построить максимальное (минимальное) остовное дерево для данного нагруженного графа (рисунок 32). Веса ребер графа взять из таблицы 1.

Таблица 1 – Веса ребер графа

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
1	3	2	4	5	4	6	4	7	3	2	5	6	7	5	6	4	8	2	7
3	2	1	5	4	3	7	6	4	5	3	6	8	5	4	3	7	8	9	5
6	4	9	1	8	5	6	7	4	3	5	2	1	9	2	5	3	6	4	7
1	3	5	7	9	4	2	4	6	3	5	7	4	2	5	3	7	4	8	3
5	3	2	6	6	7	8	4	3	5	2	8	6	4	7	4	2	9	5	6

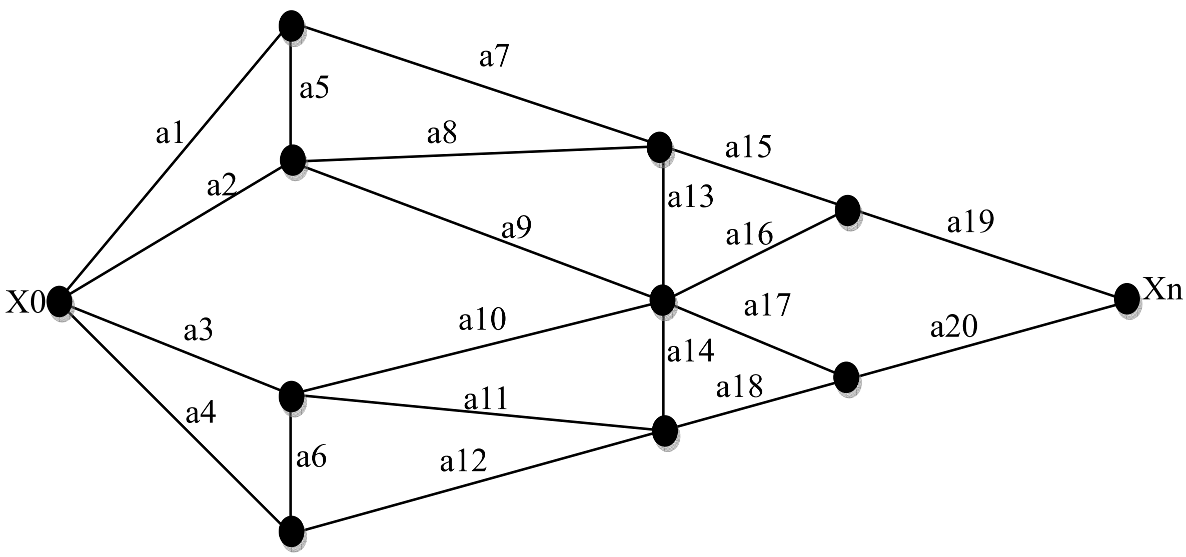


Рисунок 32

7 Элементы сетевого планирования

Методы сетевого планирования и управления (СПУ) используются при планировании сложных комплексных проектов, например, таких как:

– строительство и реконструкция каких-либо объектов;
– выполнение научно-исследовательских и конструкторских работ;
– подготовка производства к выпуску продукции;
– перевооружение армии;
– развертывание системы медицинских или профилактических мероприятий.

Характерной особенностью таких проектов является то, что они состоят из ряда отдельных, элементарных работ. Они обуславливают друг друга так, что выполнение некоторых работ не может быть начато раньше, чем завершены некоторые другие.

СПУ состоит из трех основных этапов:

1) структурное планирование;
2) календарное планирование;
3) оперативное управление.

Структурное планирование начинается с разбиения проекта на четко определенные операции, для которых определяется продолжительность. Затем строится сетевой график, который представляет взаимосвязи работ проекта. Это позволяет детально анализировать все работы и вносить улучшения в структуру проекта еще до начала его реализации.

Календарное планирование предусматривает построение календарного графика, определяющего моменты начала и окончания каждой работы и другие временные характеристики сетевого графика. Это позволяет, в частности, выявлять критические операции, которым необходимо уделять особое внимание, чтобы закончить проект в директивный срок. Во время календарного планирования определяются временные характеристики всех работ с целью проведения оптимизации сетевой модели, которая улучшает эффективность использования какого-либо ресурса.

В ходе оперативного управления используются сетевой и календарный графики для составления периодических отчетов о ходе выполнения проекта. При этом сетевая модель может подвергаться оперативной корректировке, вследствие чего будет разрабатываться новый календарный план остальной части проекта.

Основными понятиями сетевых моделей являются понятия события и работы.

Работа – это некоторый процесс, приводящий к достижению определенного результата и требующий затрат каких-либо ресурсов, имеет протяженность во времени.

Событие – момент времени, когда завершаются одни работы и начинаются другие. Событие представляет собой результат проведенных работ и, в отличие от работ, не имеет протяженности во времени.

Таким образом, начало и окончание любой работы описываются парой событий, которые называются начальным и конечным событиями. Поэтому для идентификации конкретной работы используют код работы (i, j), состоящий из номеров начального (i-го) и конечного (j-го) событий (рисунок 33).

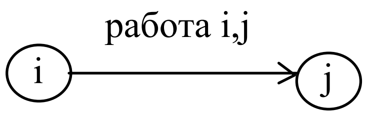


Рисунок 33 – Работа (i, j)

На этапе структурного планирования взаимосвязь работ и событий изображается с помощью сетевого графика, где работы изображаются стрелками, которые соединяют вершины, изображающие события. Работы, выходящие из некоторого события не могут начаться, пока не будут завершены все операции, входящие в это событие.

Событие, не имеющее предшествующих ему событий, т. е. с которого начинается проект, называются исходным. Событие, которое не имеет последующих событий и отражает конечную цель проекта, называется завершающим.

По данным таблицы 2 построим сетевой график (сеть).

Таблица 2 – Исходные данные задачи

События (потомки)	События (предки)				
	Начало работ	Готовность деталей	Готовность документации	Поступление дополнительного оборудования	Готовность блоков
Готовность деталей	Изготовление деталей (4/3)				
Готовность документации					Подготовка документации (5/2)
Поступление дополнительного оборудования	Закупка дополнительного оборудования (10/5)				
Готовность блоков		Сборка блоков (6/4)			
Готовность изделия			Составление инструкций (11/6)	Установка дополнительного оборудования (10/5)	Компоновка изделия (9/6)

Выбираем исходное событие (не имеющее предков) – «начало работ». После начального события следуют события: «готовность деталей» и «поступление дополнительного оборудования», которые связаны с начальным событием работами: «изготовление деталей» и «закупка дополнительного оборудования».

После события готовность деталей выполняется работа «сборка блоков» и следует событие «готовность блоков».

Продолжая рассуждения, получим следующую сеть (рисунок 34).

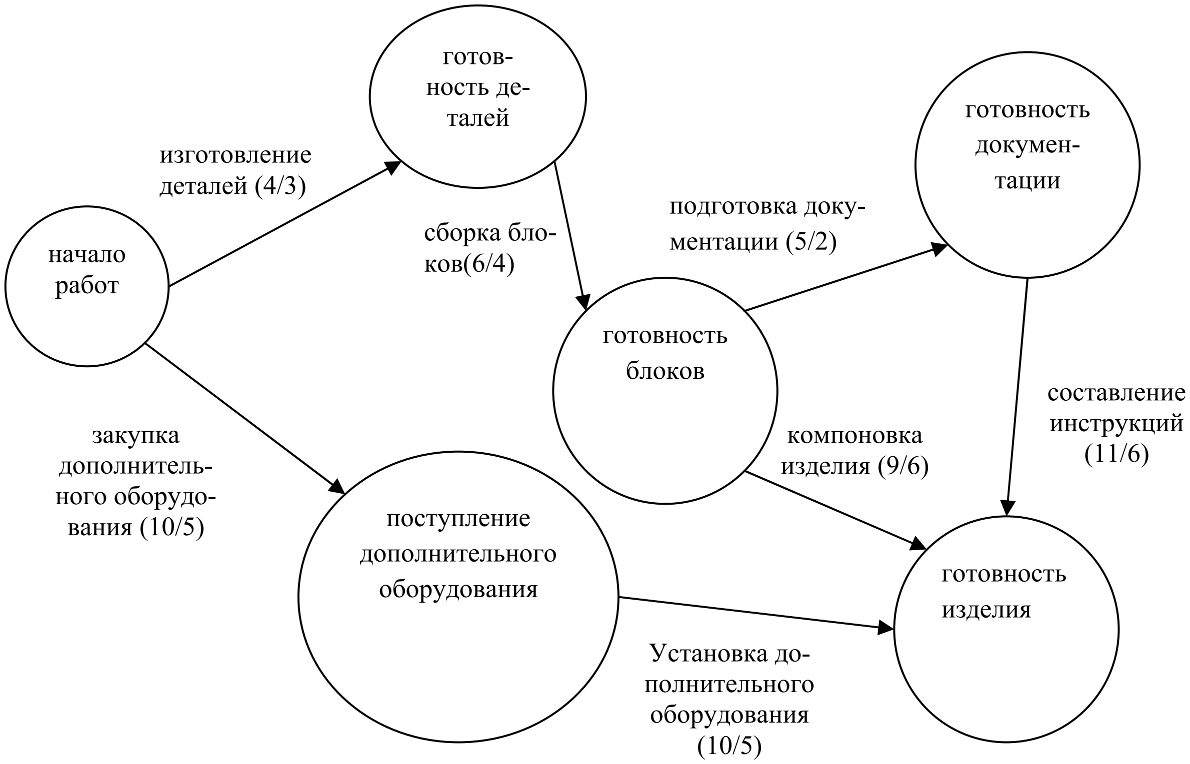


Рисунок 34 – Сетевой график

Обозначим события – получим сетевой график (рисунок 35).

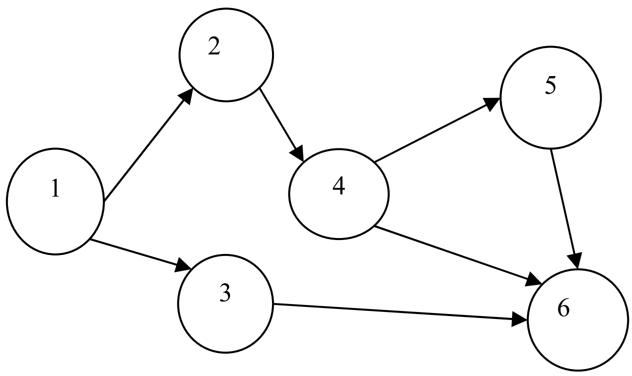


Рисунок 35 – Сетевой график

Полученный сетевой график является адекватным.

Путь – это любая последовательность работ в сетевом графике, в которой конечное событие одной работы совпадает с начальным событием следующей за ней работы.

Полный путь – это путь от исходного до завершающего события.

Критический путь – максимальный по продолжительности полный путь.

Подкритический путь – полный путь, ближайший по длительности к критическому пути.

Работы, лежащие на критическом пути, называют критическими.

Применение методов СПУ в конечном счете должно обеспечить получение календарного плана, определяющего сроки начала и окончания каждой операции. Построение сети является лишь первым шагом на пути к достижению этой цели. Вторым шагом является расчет сетевой модели, который выполняют прямо на сетевом графике, пользуясь простыми правилами.

К временным параметрам событий относятся:

– ранний срок наступления события i – $T_{\text{р}}(i)$;

– поздний срок наступления события i – $T_n(i)$;

– резерв времени наступления события i – $R(i)$.

$T_{\text{р}}(i)$ – это время, необходимое для выполнения всех работ, предшествующих данному событию *i*.

$T_n(i)$ – это такое время наступления события *i*, превышение которого вызовет аналогичную задержку наступления завершающего события сети.

$R(i)$ – это такой промежуток времен, на который может быть отсрочено наступление этого события без на рушения сроков завершения разработки в целом.

Значения временных параметров записываются прямо в вершины на сетевом графике следующим образом (рисунок 36).

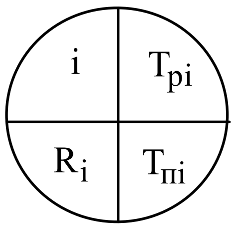


Рисунок 36 – Событие. Временные параметры

Расчет ранних сроков свершения событий ведется от исходного к завершающему событию.

1 Для исходного события $T_{\text{р}}(i) = T_n(i) = 0$.

2 Для всех остальных событий $T_{\text{р}}(i) = \max\limits_{k<i}\left[T_{\text{р}}(k) + t(k,i)\right]$, где максимум берется по всем работам (k,i), входящим в событие *i*.

Поздние сроки свершения событий рассчитываются от завершающего к исходному событию.

3 Для завершающего события $T_n(i) = T_{\text{р}}(i)$.

4 $T_n(i) = \min\limits_{j>i}\left[T_{\text{р}}(j) - t(i,j)\right]$, где минимум берется по всем работам (i,j), выходящим из события *i*.

5 $R(i) = T_n(i) - T_{\text{р}}(i)$.

На основе ранних и поздних сроков событий можно определить временные параметры работ сети (рисунок 37).

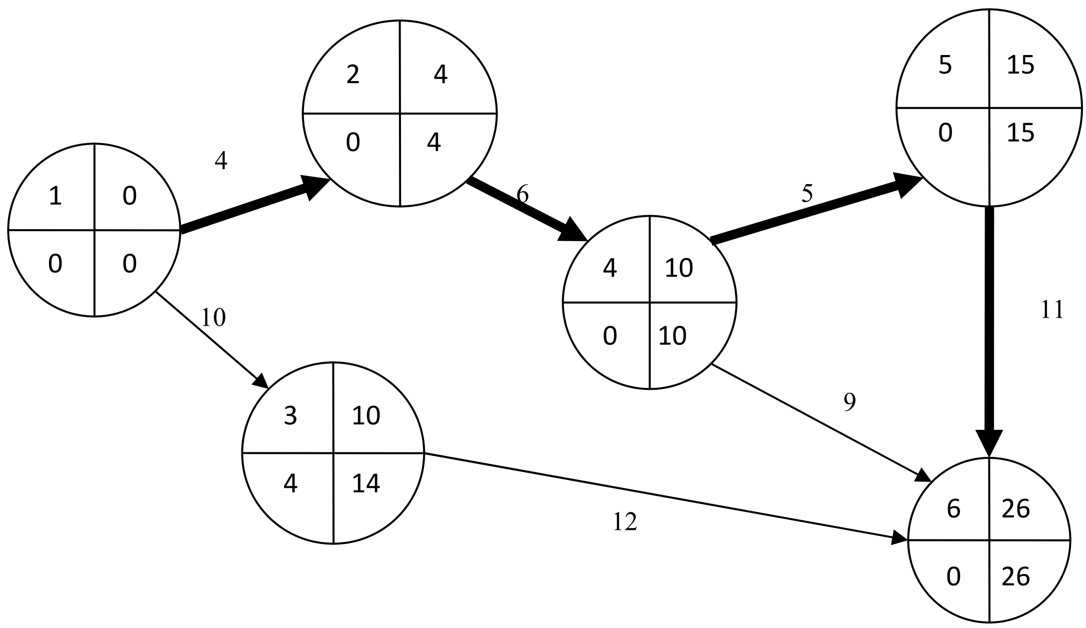


Рисунок 37 – Сетевой график

Критический путь – $1 \to 2 \to 4 \to 5 \to 6$. Критическое время $t_{кр} = 26$ суток.

Задачи для самостоятельного решения

Рассчитать непосредственно на сетевом графике комплекса работ ранние и поздние сроки свершения событий, резервы времени событий, минимальное время выполнения комплекса (критический срок). Выделить на сетевом графике критический путь. Для некритических работ найти полные и свободные резервы времени (рисунок 38).

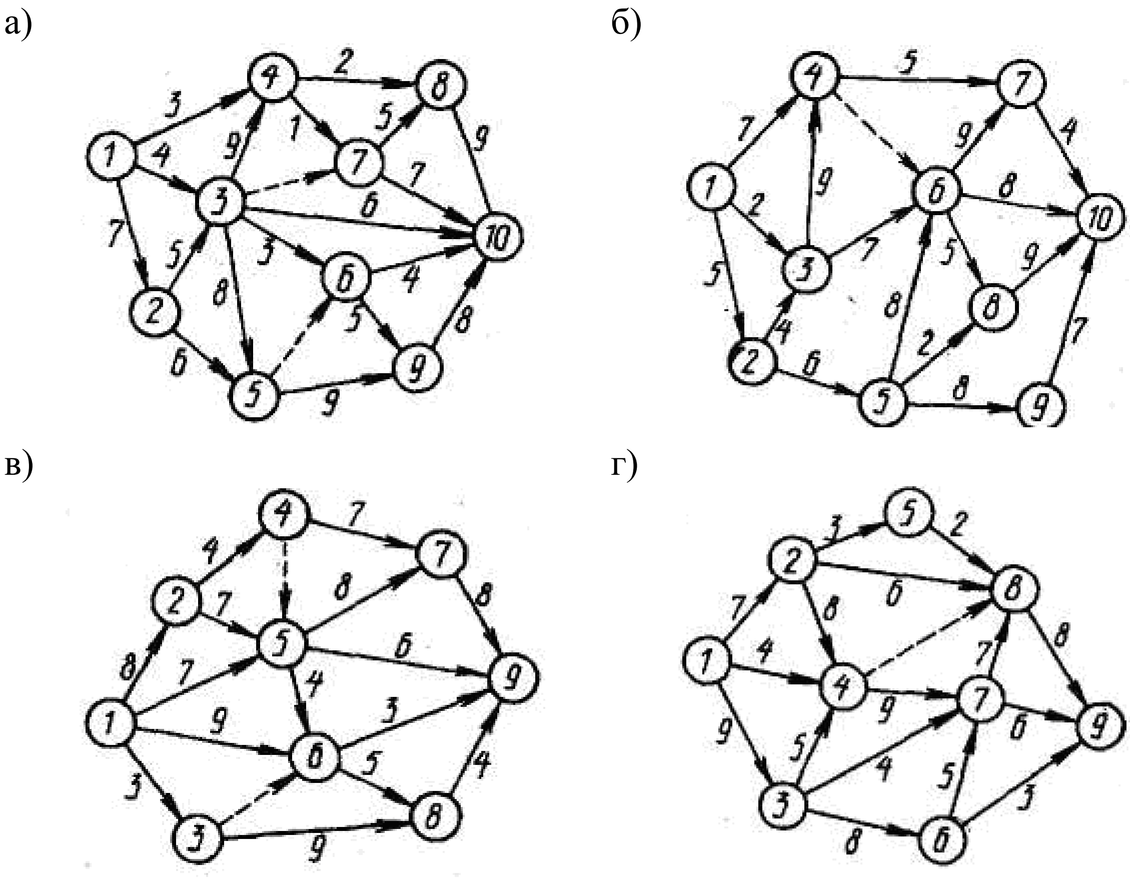


Рисунок 38

8 Потоки на сетях

Через сеть по её ребрам или дугам пропускают некоторое вещество. Вещество проходит из истока в сток по ребрам данной сети.

Наша задача состоит в расчете максимального количества вещества, которое может пропустить через себя данная сеть.

Определение: пропускной способностью ребра r_{ij} называется количество вещества, которое может пройти по данному ребру за единицу времени.

Пропускная способность ребра в прямом и обратном направлениях не одинакова. Если две вершины не соединены ребром, то пропускная способность между ними не равна нулю.

Если на графе над каждым ребром указать пропускную способность в прямом и обратном направлениях, то соответствующие элементы ее матрицы будут положительны.

Поток по ребру – это количество вещества, проходящего по ребру за единицу времени x_{ij}.

Множество всех потоков по ребру образует поток по сети: $X=\{x_{ij}\}$.

Свойства потоков:

1) $x_{ij} = x_{jj}$, то есть их пропускная способность одинакова в прямом и

обратном направлениях;

2) $x_{ij} = 0$ – поток по вершине;

3) $x_{ij} \leq r_{ij}$, r_{ij} – пропускная способность ребра.

Если $x_{ij} < r_{ij}$, то ребро ненасыщенное; если $x_{ij} = r_{ij}$, то ребро насыщенное.

Вершина, с которой рассматривается поток, называется истоком, а в котором заканчивается стоком.

Теорема Форда- Фалкерсона: максимальный поток по заданной сети равен минимальной пропускной способности разреза сети.

Разобьем множество сети на подмножество X и $\overline{X}$ (не X).

Разрез – множество дуг, для которых выполняются следующие требования:

1) $I \in X$, – исток; $S \in \overline{X}$ – сток;

2) $X \bigcap \overline{X} = \varnothing$;

3) $X \bigcup \overline{X} = N$, где N – множество всех вершин в сети.

Пропускная способность разреза – это сумма пропускных способностей ребер, входящих в разрез.

Задача о максимальном потоке решается следующим образом.

Дан некоторый поток, который направлен из истока в сток.

Разобьём сеть на два подмножества:

– к первому отнесём все вершины, достигаемые из истока, по ненасыщенным рёбрам;

– ко вторым отнесём все оставшиеся вершины.

При таком разбиении возможны два случая:

1) исток и сток находятся в разных подмножествах, тогда задача решена;

2) исток и сток находятся в первом подмножестве, требуется каким-либо методом увеличить поток из истока в сток и заново разбить сеть на два подмножества.

Расчет непосредственно на сети:

1) выписываем полные пути из истока в сток, считая при этом пропускную способность. Пропускная способность сети равна минимальной пропускной способности.

Над каждым ребром пути записываем поток по данному пути. Если поток и пропускная способность совпадают, то ребро насыщенное;

2) строим другие полные пути и считаем их пропускную способность. Если какое-либо ребро пути входит в другой путь, то пропускная способность данного ребра равна первоначальной пропускной способности минус поток по другому пути. К потоку данного ребра прибавляем поток по новому пути. Выделяем все насыщенные ребра данного пути;

3) строим следующий полный путь до тех пор, пока сток достигает истока по не насыщенному пути;

4) выделяем разрез на сети;

5) считаем пропускную способность разреза, выписываем ребра, входящие в разрез, выписываем два подмножества вершин X и $\overline{X}$.

Пример (рисунок 39).

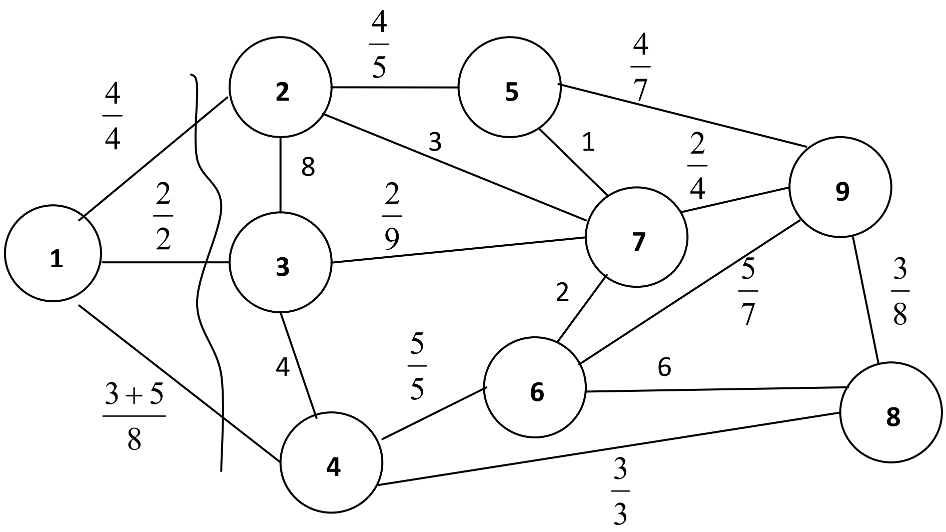


Рисунок 39

L_1: 1-2-5-9, $R(L_1)=\min\{4,5,7\}=4$; L_2:1-4-8-9, $R(L_2)=3$;
L_3: 1-3-7-9, $R(L_3)=2$; L_4: 1-4-6-9, $R(L_4)=5$.
$X=\{1\}$ $\overline{X}=\{2,3,4,5,6,7,8,9\}$; $X/\overline{X}=\{(1,2),(1,3),(1,4)\}$ (рисунок 39).
$R(X/\overline{X})=14$ ед. $R(L_1)+R(L_2)+R(L_3)+R(L_4)=14$.

Задачи для самостоятельного решения

1 На заданной сети указаны пропускные способности ребер. Предполагается, что пропускные способности в обоих направлениях одинаковы (рисунок 40). Требуется:

1) сформировать на сети поток максимальной мощности, направленный из истока I в сток S;

2) выписать ребра, образующие на сети разрез минимальной пропускной способности.

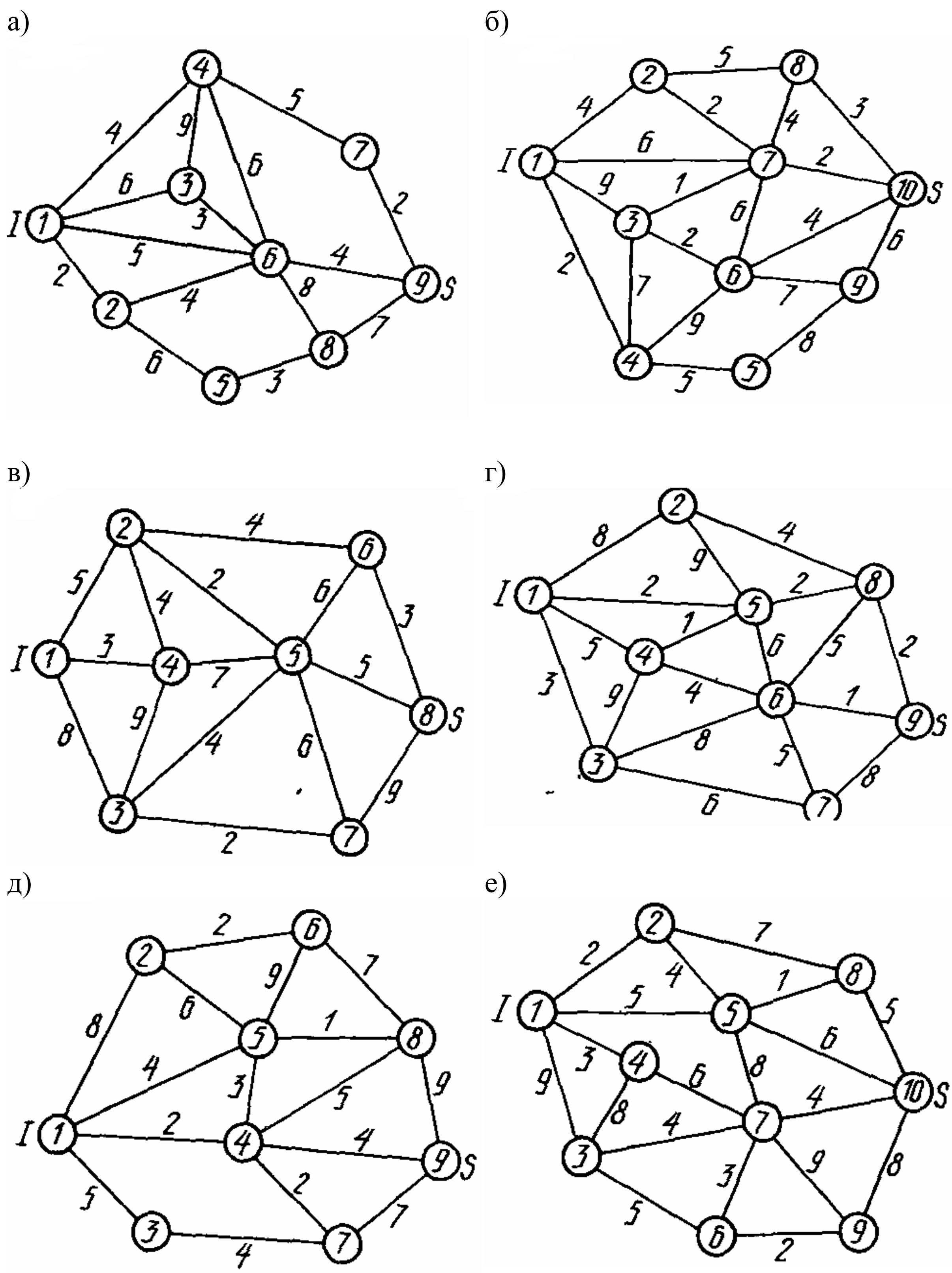


Рисунок 40

Список литературы

1 **Белоусов, А. И.** Дискретная математика : учеб. пособие для втузов / А. И. Белоусов, С. Б. Ткачев ; под ред. В. С. Зарубина и А. П. Крищенко. – 2-е изд. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. – 416 с.

2 Сборник задач и упражнений по высшей математике: Математическое программирование : учеб. пособие / А. В. Кузнецов [и др.] ; под общ. ред. А. В. Кузнецова. – Минск : Выш. шк., 2002. – 447 с.

3 **Горбатов, В. А.** Основы дискретной математики : учеб. пособие / В. А. Горбатов. – М. : Высш. шк., 1986. – 311 с.

4 **Кузин, Л. Т.** Основы кибернетики : учеб. пособие / Л. Т. Кузин. – Т. 1: Математические основы кибернетики. – М. : Энергия, 1973. – 354 с.

5 **Коршунов, Ю. М.** Математические основы кибернетики : учеб. пособие / Ю. М. Коршунов. – М. : Энергия, 1972. – 475 с.

6 **Нефедов, В. И.** Курс дискретной математики : учеб. пособие / В. И. Нефедов, В. А. Осипова. – М. : Изд-во МАИ, 1992. – 446 с.

7 **Белов, В. В.** Теория графов : учеб. пособие / В. В. Белов. – М. : Высш. шк., 1976. – 445 с.