

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра программного обеспечения автоматизированных систем

## **АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ**

Методические указания  
к выполнению лабораторных и курсовых работ  
для студентов направления (специальности)  
231000.62 – «Программная инженерия»

Курган 2012

Кафедра: **«Программное обеспечение автоматизированных систем»**

Дисциплина: **«Программная инженерия» (231000.62)**

Составил: канд. техн. наук, доцент А.М. Семахин

Утверждены на заседании кафедры «3» июня 2011 г.

Рекомендованы методическим советом университета «12» октября 2012 г.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. Линейные структуры данных.....	5
1.1. Линейный список .....	5
1.2. Стек .....	6
1.3. Очереди.....	6
1.3.1. Универсальная очередь неограниченного размера .....	6
1.3.2. Универсальная очередь ограниченного размера .....	7
2. Нелинейные структуры данных. Бинарные деревья.....	7
3. Алгоритмы сортировки данных в оперативной памяти .....	7
3.1. Сортировка массива простым выбором .....	8
3.2. Сортировка массива вставками (сортировка Шелла) .....	8
3.3. Сортировка массива обменом (сортировка Хоара).....	8
3.4. Генерация массива числовых данных случайным распределением значений элементов .....	9
4. Внутренний поиск данных в таблице.....	11
4.1. Последовательный поиск.....	11
4.2. Логарифмический (бинарный) поиск .....	12
4.3. Поиск с использованием перемешанной таблицы (хэш-таблицы).....	12
5. Алгоритмы поиска на графах.....	12
5.1. Поиск кратчайшего пути. Алгоритм Дейкстры.....	12
5.2. Поиск кратчайшего пути. Алгоритм Флойда .....	15
6. Задания для выполнения лабораторных работ.....	21
6.1. Задания для выполнения лабораторной работы №1 «Линейные структуры данных» .....	21
6.2. Задания для выполнения лабораторной работы №2 «Нелинейные структуры данных. Бинарные деревья».....	27
6.3. Задания для выполнения лабораторной работы №3 «Алгоритмы сортировки данных в оперативной памяти» .....	34
6.4. Задания для выполнения лабораторной работы №4 «Внутренний поиск данных в таблице» .....	35
6.5. Задания для выполнения лабораторной работы №5 «Алгоритмы поиска на графах» .....	40
6.5.1. Поиск кратчайшего пути. Алгоритм Дейкстры .....	40
6.5.2. Поиск кратчайшего пути. Алгоритм Флойда .....	49
7. Курсовая работа.....	59
7.3.1. Назначение, цели и задачи курсовой работы .....	59
7.3.2. Требования к курсовой работе.....	59
7.3.2.1. Требования к функциональным характеристикам .....	59
7.3.2.2. Требования к эксплуатационным характеристикам.....	59
7.3.2.3. Требования к программному обеспечению .....	60
7.3.2.4. Требования к содержанию курсовой работы .....	60
7.3.3. Варианты заданий курсовой работы.....	60
ЗАКЛЮЧЕНИЕ .....	62
СПИСОК ЛИТЕРАТУРЫ.....	63

## ВВЕДЕНИЕ

Организация данных определяет алгоритм работы программы. Выбор структуры данных – важный этап разработки программы. Структуры данных разделяются на динамические структуры и структуры фиксированного размера, линейные и нелинейные.

Структуры данных фиксированного размера – массивы, структуры. Память под данные выделяется на этапе компиляции. Необходимый объем известен до начала выполнения программы и задан в виде константы.

Динамические структуры – линейные списки, стеки, очереди и бинарные деревья. Память выделяется во время выполнения программы с помощью операций `new` или функций `malloc()`, `calloc()`. Объем должен быть известен до распределения памяти.

Динамические структуры данных различаются способами связи элементов и допустимыми операциями и занимают несмежные участки оперативной памяти.

Линейные структуры данных – списки, стеки, очереди. Нелинейные структуры – бинарные деревья.

Связанный список – структура данных, состоящая из единиц данных, выстроенных в цепочку. Вставка и удаление производится в любом месте связанного списка.

Стек – структура данных, вставка и удаление данных производится в вершине.

Очередь – структура данных, связанная с ожиданием. Вставка производится в конце очереди (хвост), а удаление – в начале очереди (голова).

Бинарные деревья – структура данных, состоящая из узлов, каждый из которых содержит не более двух ссылок на бинарные деревья.

Элемент динамической структуры представляет структуру, содержащую два поля: для хранения данных и указателя. Полей данных и указателей может быть несколько. Поля данных могут быть любого типа: основного, составного или типа указатель.

Динамические структуры применяют для эффективной работы с данными: сортировка и поиск данных, устранение дубликатов, представление каталогов файловых систем и компиляция выражений в машинный язык. Например, при упорядочивании массива данных организуется линейный список. Для поиска элемента данные представляются в виде бинарного дерева.

В методических указаниях приведен теоретический материал по разделам и представлены варианты заданий по каждой теме для выполнения практических работ.

Формализация структур данных производится структурным и объектно-ориентированным методами программирования.

# 1. Линейные структуры данных

## 1.1. Линейный список

Линейные структуры данных – связанный список, стек и очередь.

Связанный список – набор объектов автореферентного класса, называемых узлами, соединенных указателями-связками.

Автореферентный класс – класс, содержащий элемент-указатель, ссылающийся на объект того же классового типа.

Доступ к связанному списку осуществляется через указатель (ключ) на первый узел списка, обращение к последующему узлу осуществляется через указатель, хранящийся в предыдущем узле. Указатель-связка в последнем узле списка устанавливается в нуль, который отмечает конец списка. Данные сохраняются динамически – узлы создаются по мере необходимости. Узел может содержать данные любого типа, включая объекты других классов. Если узлы содержат указатели базового класса на объекты базового или производного классов, связанных отношением наследования, то можно создавать из таких узлов связанный список и полиморфно обрабатывать объекты посредством вызова виртуальных функций.

Над списками выполняются операции:

- формирование списка (создание первого элемента);
- добавление элемента в конец списка;
- чтение элемента с заданным ключом;
- вставка элемента в заданное место списка (до или после элемента с заданным ключом);
- удаление элемента с заданным ключом;
- упорядочивание списка по ключу.

Способы реализации списков:

- односвязный список – список начинается с указателя на первый узел. Узел содержит указатель на последующий узел. Заканчивается указателем, имеющим значение нуль. Односвязный список можно проходить в одном направлении;

- циклический односвязный список – узел начинается с указателя на первый узел. Каждый узел содержит указатель на следующий узел. Последний узел не содержит нулевого указателя и ссылается на первый узел;

- двусвязный список – список, включающий два начальных указателя. Один указывает на первый элемент списка и позволяет пройти список от начала к концу, а второй указывает на последний элемент и позволяет пройти список от конца к началу. Каждый узел имеет прямой указатель на следующий узел списка в прямом направлении и обратный указатель на следующий узел в обратном направлении;

- мультисписки – структуры данных, состоящие из элементов, которые принадлежат нескольким спискам, не повторяясь в нескольких экземплярах. Элементы имеют указатели для каждого списка, которому они принадлежат.

Списки могут быть линейными и циклическими, однонаправленными и двунаправленными.

Связные списки имеют преимущества перед массивами:

- 1) связанные списки являются динамическими;
- 2) связанные списки заполняются до отказа, когда памяти в системе недостаточно для удовлетворения запросов динамического выделения памяти;
- 3) связанный список производит эффективную операцию вставки/удаления в любом месте списка [1; 2; 3].

## 1.2. Стек

Стек – частный случай однонаправленного списка, добавление элементов и выборка выполняются с одного конца, называемого вершиной стека. Другие операции со стеком не определены. При выборе элемент исключается из стека. Стек реализует принцип обслуживания LIFO (last in – first out, последний пришел – первым ушел). Стеки применяются в системном программном обеспечении, компиляторах, в рекурсивных алгоритмах.

## 1.3. Очереди

Очередь – частный случай однонаправленного списка. Добавление элементов выполняется в один конец, а выборка – из другого конца. Другие операции с очередью не определены. При выборке элемент исключается из очереди. Очередь реализует принцип обслуживания FIFO (first in - first out, первым пришел – первым ушел).

Над очередью определены три операции:

- занесение элемента в очередь;
- выбор элемента из очереди;
- просмотр элементов очереди.

При выборе элемента из очереди он исключается. В очереди доступны левый и правый конец, в которые можно заносить или выбирать элементы.

В программировании очередь применяется при моделировании, диспетчеризации задач операционной системой, буферизованном вводе/выводе.

### *1.3.1. Универсальная очередь неограниченного размера*

Универсальная очередь неограниченного размера строится на базе двунаправленного линейного списка с размещением элементов в динамической памяти и не имеет ограничений по размеру. Недостаток очереди неограниченного размера – повышенный расход памяти.

### **1.3.2. Универсальная очередь ограниченного размера**

Универсальная очередь ограниченного размера реализуется на базе массива. Для каждого элемента очереди хранится последняя информация. Связи между элементами универсальной очереди ограниченного размера и указателями на концы очереди определяются индексами массива [4; 5; 6].

## **2. Нелинейные структуры данных. Бинарные деревья**

Бинарные деревья – динамическая структура данных, состоящая из узлов, каждый из которых содержит данные и не более двух ссылок на бинарные деревья. На каждый узел имеется одна ссылка. Корень дерева – начальный узел дерева. Лист дерева – узел, не имеющий поддеревьев. Предки – исходящие узлы, потомки – входящие узлы. Высота дерева – количество уровней, на которых располагаются узлы.

Дерево поиска – дерево, у которого для каждого узла все ключи левого поддерева меньше ключа этого узла, а все ключи правого поддерева больше ключа этого узла. В дереве поиска можно найти элемент по ключу, двигаясь от корня и переходя на левое или правое поддерево в зависимости от значения ключа в каждом узле. Поиск по ключу эффективнее поиска по списку, поскольку время поиска определяется высотой дерева. Высота дерева пропорциональна двоичному логарифму количества узлов.

Дерево является рекурсивной структурой данных. Действия с деревьями описываются рекурсивными алгоритмами. Деревья поиска применяются для сортировки значений. При обходе дерева узлы не удаляются.

Для бинарных деревьев определены операции:

- включение узла дерева;
- поиск по дереву;
- обход дерева;
- удаление узла.

## **3. Алгоритмы сортировки данных в оперативной памяти**

Сортировка – процесс перестановки объектов множества в определенном порядке. Цель сортировки – упрощение поиска элементов в отсортированном множестве. Методы сортировки разделяют на две категории:

- сортировка массивов (внутренняя сортировка);
- сортировка последовательных файлов (внешняя сортировка).

Основное требование к методам сортировки массивов – экономное использование памяти: сортировку нужно выполнять «на том же месте» (in site).

Методы сортировки подразделяются на три класса:

- Сортировка выбором. Выбирается элемент с наибольшим значением ключа и меняется местами с последним. Затем то же повторяется для s-1

элемента. Найденный элемент с наибольшим значением ключа меняется местами с предпоследним элементом и т.д.

- Сортировка вставками. Элементы разделяются на готовую последовательность (упорядоченную) и неупорядоченную. В начале упорядоченная часть содержит один элемент. Очередной элемент из неупорядоченной части вставляется на подходящее место в упорядоченную часть. Упорядоченная часть удлиняется на один элемент, а неупорядоченная укорачивается. Сортировка заканчивается при исчезновении неупорядоченной части.

- Сортировка обменом. Выполняется обмен местами двух соседних элементов (перестановка), если они расположены не так, как требует отсортированный массив.

### **3.1. Сортировка массива простым выбором**

Алгоритм сортировки включает этапы:

1. Выбирается элемент с наибольшим значением ключа.
2. Элемент меняется местами с последним элементом  $arr[s-1]$ . Операции повторяются с оставшимися первыми  $s-1$  элементами, затем – с  $s-2$  первыми элементами и т.д. до тех пор, пока не останется только один первый элемент – наименьший.

### **3.2. Сортировка массива вставками (сортировка Шелла)**

Алгоритм сортировки включает этапы:

1. Используется несколько проходов.
2. На первом проходе отдельно группируются и сортируются вставками элементы, отстоящие друг от друга на  $d=size/2$  позиций.
3. На втором проходе группируются и сортируются вставками элементы, отстоящие друг от друга на  $d=d/2$  позиций.
4. Аналогично выполняются следующие проходы. Сортировка заканчивается последним проходом при  $d=1$  (как при простой сортировке вставками).

### **3.3. Сортировка массива обменом (сортировка Хоара)**

Алгоритм сортировки включает этапы:

1. Исходный массив разбивается на два сегмента - левый, элементы которого  $arr[i].key \leq median.key$ , и правый сегмент с элементами  $arr[j].key \geq median.key$ .
2. Каждый из полученных сегментов сортируется автономно, аналогично предыдущему, до получения сегментов единичной длины [5; 6].



### 3.4. Генерация массива числовых данных случайным распределением значений элементов

Генератор случайных чисел использует функции rand(), srand().

Программа, реализующая получение последовательности случайных чисел, равномерно распределенных на отрезке [0;1], приведена в листинге 1.

Листинг 1. Реализация равномерного распределения.

```
#include<stdlib>
#include <ctime>
using name space std;
const int N=1000; // Длина случайной последовательности
srand((unsigned)time(0));
for (int i=0; i<N; i++)
r=get_uniform();
.....
Float get_uniform()
{
int number=rand();
float result=(float)number/(RAND_MAX+1);
return(result);
}
```

В табл. 3.1 перечислены вероятностные распределения, для которых обратная функция  $F^{-1}(u)$  является элементарной.

Таблица 3.1

Распределение вероятностей с элементарной функцией

N п/п	Название	Функция F(x)	Функция $F^{-1}(u)$
1	Bradford	$\frac{\ln(1 + \frac{C * (x - A)}{B - A})}{\ln(C + 1)},$ $A < x < B, C > 0$	$A + \frac{(B - A) * ((C + 1)^u - 1)}{C}$
2	Burr	$(1 + (\frac{x - A}{B})^{-C})^{-D-1},$ $y > A, B > 0, C > 0, D \leq 100$	$A + B * (u^{\frac{-1}{D+1}} - 1)^{\frac{-1}{C}}$
3	Cauchy	$\frac{1}{2} + \frac{1}{\pi} * \frac{1}{\text{tg}(\frac{x - A}{B})}, B > 0$	$A + B * \text{arctg}(\frac{2}{\pi * (2 * u - 1)})$
4	Exponential	$1 - e^{-\frac{A-x}{B}}, x \geq A, B > 0$	$A - B * \ln(1 - u)$

5	ExtremeL B	$\exp(-(\frac{x-A}{B})^{-C}),$ $x > A, B > 0, 0 < C \leq 100$	$A + B * (-\ln u)^{\frac{-1}{C}}$
6	Fisk	$\frac{1}{1 + (\frac{x-A}{B})^{-C}},$ $x > A, B > 0, 0 < C \leq 100$	$A + B * (\frac{1}{u} - 1)^{\frac{-1}{C}}$
7	Gumbel	$\exp(-\exp(\frac{A-x}{B})), B > 0$	$A - B * \ln(-\ln u)$
8	Laplace	$\begin{cases} \frac{1}{2} * \exp(\frac{x-A}{B}), x \leq A; \\ 1 - \frac{1}{2} * \exp(\frac{A-x}{B}), x > A \end{cases}$	$\begin{cases} A + B * \ln(2 * u), u < 0.5; \\ A - B * \ln(2 * (1-u)), u > 0 \end{cases}$
9	Logistic	$\frac{1}{1 + \exp(\frac{A-x}{B})}, B > 0$	$A - B * \ln(\frac{1}{u} - 1)$
10	Pareto	$1 - (\frac{A}{x})^B, 0 < A \leq x, B > 0$	$A * (1-u)^{\frac{-1}{B}}$
11	Reciproca 1	$\ln(\frac{A}{x}) / \ln(\frac{A}{B}),$ $0 < A \leq x \leq B$	$B^u * A^{1-u}$
12	Weibull	$1 - \exp(-(\frac{x-A}{B})^C),$ $x > A, B > 0, C > 0$	$A + B * (-\ln(1-u))^{\frac{1}{C}}$

Программа, реализующая получение последовательности случайных чисел, распределенных по вероятностному закону Парето (листинг 2).

Листинг 2. Реализация вероятностного закона распределения Парето.

```
#include <stdio.h>
#include <cmath>
#include <cstdlib>
using namespace std;
const int VOLUME=1000; //Количество элементов случайной
последовательности
/* Прототип ГСЧ. А и В – параметры распределения Парето. А>0, В>0 */
float get_pareto(float A, float B);
int main() {
```

```

FILE *out; //Указатель выходного файла
float A, B, result;
out=fopen("out_pareto", "wt");
//Ввод параметров
printf("\nA="); scanf("%f", &A);
printf("\nB="); scanf("%f", &B);
srand((unsigned)time(0)); //Инициализация датчика равномерного
распределения
for (int i=0; i<VOLUME; i++)
{ /* Вызов функции возвращает случайное число, полученное по
вероятностному закону распределения Парето*/
result=get_pareto(A, B);
fprintf(out, "%f\n", result);}
fclose(out);
return 0;
}
float get_pareto(float A, float B)
{ int r_num; float root, right;
r_num=rand(); /* получение случайного целого числа */
right=(float)r_num/RAND_MAX+1; /* Проекция на интервал (0,1) */
root=A/(pow(1-right,1.0/B)); /* Вычисление значения обратной функции
*/
return (root);
} [7].

```

## 4. Внутренний поиск данных в таблице

Поиск данных подразделяется на два вида:

1. Внутренний поиск – поиск в оперативной памяти, в таблице (массиве).
2. Внешний поиск – поиск данных на внешней памяти (магнитном диске).

Таблица данных располагается в оперативной памяти и содержит некоторое количество строк. Строка занимает определенное количество байт. Байт хранит код символа.

Методы поиска в таблице подразделяются на:

1. Последовательный поиск.
2. Логарифмический поиск (бинарный, с помощью двоичного дерева).
3. Поиск, использующий прямой доступ к таблице.
4. Поиск с использованием перемешанной, слабо заполненной таблицы (хэш-таблицы).

### 4.1. Последовательный поиск

Поиск выполняется в полностью заполненной таблице. Просмотр таблицы выполняется последовательно, в соответствии с ростом строк таблицы. Если в какой-то строке таблицы поле ключа совпадает с ключевым словом, то

поиск окончен с результатом «найдено». Если этого не произошло, а конец таблицы достигнут, то поиск окончен с результатом «не найдено».

## **4.2. Логарифмический (бинарный) поиск**

Логарифмический поиск включает два этапа:

1. Начальная подготовка таблицы в форме двоичного дерева.
2. Бинарный (логарифмический) поиск в подготовленной таблице.

Исходная таблица предварительно подготавливается в форме двоичного дерева так, чтобы ключи левого поддерева были раньше по алфавиту, чем ключ корня, а ключи правого поддерева позже. Область поиска сокращается в два раза.

Бинарный поиск включает шаги:

1. Исходный ключ сравнивается с ключом, соответствующим корню дерева. Если ключи совпадают, то нужная строка найдена и поиск завершен.

2. Иначе определяется поддерево для продолжения поиска путем сравнения ключевого слова с ключом таблицы. Если  $KeyWord < pTable[i-1].key$ , то необходимо вести поиск в левом поддереве и номер следующей вершины  $i=i*2$ . При выполнении противоположного условия  $KeyWord > pTable[i-1].key$  поиск следует вести в правом поддереве и номер следующей вершины дерева  $i=i*2+1$ .

3. При выполнении условия  $i > size$  (дерево не содержит вершину  $i$ ) поиск следует прекратить, так как строка с ключевым словом  $KeyWord$  отсутствует. Иначе выполняется переход к шагу 1 с новым значением  $i$ , соответствующим корню левого или правого поддерева.

## **4.3. Поиск с использованием перемешанной таблицы (хэш-таблицы)**

При поиске с использованием хэш-таблицы применяется организация данных в виде массива. Сущность поиска заключается в преобразовании заданного ключа  $KeyWord$  в индекс  $Hash(KeyWord)$  соответствующей строки в таблице (поиск с преобразованием ключей).

Поиск с использованием перемешанной таблицы включает два этапа:

1. Вычисление соответствующего индекса  $Hash(KeyWord)$  в таблице.
2. Проверка наличия элемента с ключом  $KeyWord$  в таблице в строке с индексом  $Hash(KeyWord)$  [9; 10; 11; 12].

## **5. Алгоритмы поиска на графах**

### **5.1. Поиск кратчайшего пути. Алгоритм Дейкстры**

Алгоритм Дейкстры разработан для нахождения кратчайшего пути между заданным исходным узлом и любым другим узлом сети.

В процессе выполнения алгоритма Дейкстры используется процедура пометки ребер.

Пусть  $i$  - предыдущий узел графа,  $j$  – последующий узел,  $S_i$  - кратчайшее расстояние от исходного узла 1 до узла  $i$ ,  $l_{ij}$  длина ребра графа  $(i,j)$ .

Метка  $[S_i, i]$  для узла  $j$  определяется следующим образом

$$[S_j, j] = [S_i + l_{ij}, j], \quad l_{ij} \geq 0$$

Метки в алгоритме Дейкстры могут быть двух типов: временные и постоянные.

Временная метка может быть заменена на другую временную, если будет найден более короткий путь к узлу. Статус временной метки изменяется на постоянный, если не существует более короткого пути от исходного узла к данному узлу.

Алгоритм Дейкстры включает этапы:

**Этап 1.** Исходному узлу (узел 1) присваивается метка  $[0, -]$ . Полагаем  $i=1$ .

**Этап 2.** Вычисляются временные метки  $[S_i + l_{ij}, j]$  для всех узлов  $j$ , которые можно достичь непосредственно из узла  $i$  и которые не имеют постоянных меток. Если узел  $j$  имеет метку  $[S_j, k]$ , полученную от другого узла  $k$ , и если  $S_i + l_{ij} < S_j$ , тогда метка  $[S_j, k]$  заменяется на  $[S_i + l_{ij}, j]$ .

**Этап 3.** Если все узлы имеют постоянные метки, процесс вычислений заканчивается. Иначе выбирается метка  $[S_r, r]$  с наименьшим значением расстояния  $S_r$  среди всех временных меток. Если таких меток несколько, то выбор произволен. Полагаем  $i=r$  и переходим к этапу 2.

Пример нахождения кратчайшего пути по алгоритму Дейкстры.

Постановка задачи. Транспортная сеть состоит из пяти городов (рис. 5.1). Расстояния между городами в километрах приведены возле соответствующих дуг сети. Найти кратчайшие расстояния от города 1 (узел 1) до остальных четырех городов.

Решение задачи.

1. Назначаем узлу 1 постоянную метку  $[0, -]$ .

2. Из узла 1 можно достичь узлов 2 и 3. Вычисляем метки для этих узлов, в результате получается таблица меток (табл. 5.1).

Среди узлов 2 и 3 узел 3 имеет наименьшее значение расстояния ( $u_3=30$ ).

Поэтому статус метки этого узла изменяется на «постоянная».

3. Из узла 3 можно попасть в узлы 4 и 5. Получаем следующий список узлов (табл. 5.2).

Таблица 5.1

Расчет меток для узлов 2 и 3 из узла 1

Узел	Метка	Статус
<b>1</b>	<b>[0, -]</b>	<b>Постоянная</b>
2	$[0+100, 1]=[100, 1]$	Временная
3	$[0+30, 1]=[30, 1]$	← Временная

Таблица 5.2

Расчет меток для узлов 4 и 5 из узла 3

Узел	Статус	Метка
<b>1</b>	<b>[0, -]</b>	<b>Постоянная</b>
2	[100, 1]	Временная
<b>3</b>	<b>[30, 1]</b>	<b>Постоянная</b>
4	[30+10, 3]=[40, 3]	←Временная
5	[30+60, 3]=[90, 3]	Временная

Временный статус метки [40, 3] узла 4 заменяется на постоянный ( $u_4=40$ ).

4. Из узла 4 можно достичь узлов 2 и 5. После вычисления меток получим следующий список (табл. 5.3).

Таблица 5.3

Расчет меток для узлов 4 и 5 из узла 4

Узел	Статус	Метка
<b>1</b>	<b>[0, -]</b>	<b>Постоянная</b>
2	[40+15, 4]=[55, 4]	←Временная
<b>3</b>	<b>[30, 1]</b>	<b>Постоянная</b>
<b>4</b>	<b>[40, 3]</b>	<b>Постоянная</b>
5	[90, 3] или [40+50, 4]=[90, 3]	Временная

Временная метка [100,1], полученная узлом 2 на втором шаге, изменена на [55,4]. Это указывает на то, что найден более короткий путь к этому узлу (проходящий через узел 4). Узел 5 получает две метки с одинаковым значением расстояния  $S_r=90$ .

5. Из узла 2 можно перейти только в узел 3, но он имеет постоянную метку, которую нельзя изменить. Поэтому на данном шаге получаем список меток, как и на предыдущем шаге, но с единственным изменением: метка узла 2 получает статус постоянной. С временной меткой остается только узел 5, но так как из этого узла нельзя попасть ни в какой другой, процесс вычислений заканчивается.

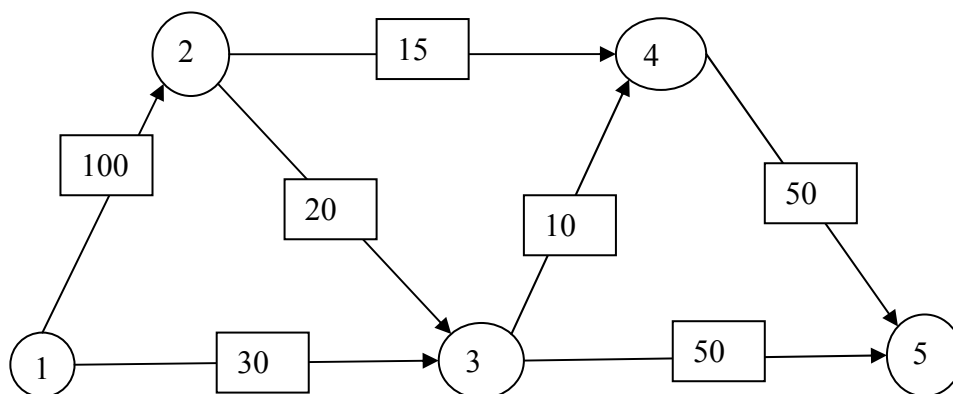


Рис. 5.1. Транспортная сеть, состоящая из 5 городов

## 5.2. Поиск кратчайшего пути. Алгоритм Флойда

Алгоритм Флойда находит кратчайшие пути между любыми двумя узлами сети. Сеть представлена в виде квадратной матрицы с  $n$  строками и  $n$  столбцами. Элемент  $(i,j)$  равен расстоянию  $d_{ij}$  от узла  $i$  к узлу  $j$ , которое имеет конечное значение, если существует дуга  $(i,j)$ , и равен бесконечности в противном случае.

Пусть даны три узла  $i, j$  и  $k$ . Заданы расстояния между ними  $d_{ij}, d_{jj}, d_{ik}$  (рис. 5.2). Если выполняется неравенство  $d_{ij} + d_{jk} < d_{ik}$ , то путь  $i \rightarrow k$  заменяется путем  $i \rightarrow j \rightarrow k$ . Такая замена называется треугольным оператором и выполняется систематически в процессе выполнения алгоритма Флойда.

Алгоритм Флойда включает этапы:

**Этап 1.** Определяем начальную матрицу расстояний  $D_0$  и матрицу последовательности узлов  $S_0$ . Диагональные элементы обеих матриц помечаются знаком  $-$ , показывающим, что эти элементы в вычислениях не участвуют. Полагаем  $k=1$ .

**Этап 2.** Задаем строку  $k$  и столбец  $k$  как ведущую строку и ведущий столбец. Рассматриваем возможность применения треугольного оператора ко всем элементам  $d_{ij}$  матрицы  $D_{k-1}$ . Если выполняется неравенство  $d_{ij} + d_{jk} < d_{ik}$ , ( $i \neq k, j \neq k, i \neq j$ ) тогда выполняются следующие действия:

2.1. Создаем матрицу  $D_k$ , путем замены в матрице  $D_{k-1}$  элемента  $d_{ij}$  на сумму  $d_{ik} + d_{kj}$ .

2.2. Создаем матрицу  $S_k$  путем замены в матрице  $S_{k-1}$  элемента  $s_{ij}$  на  $k$ . Полагаем  $k=k+1$  и повторяем шаг  $k$ .

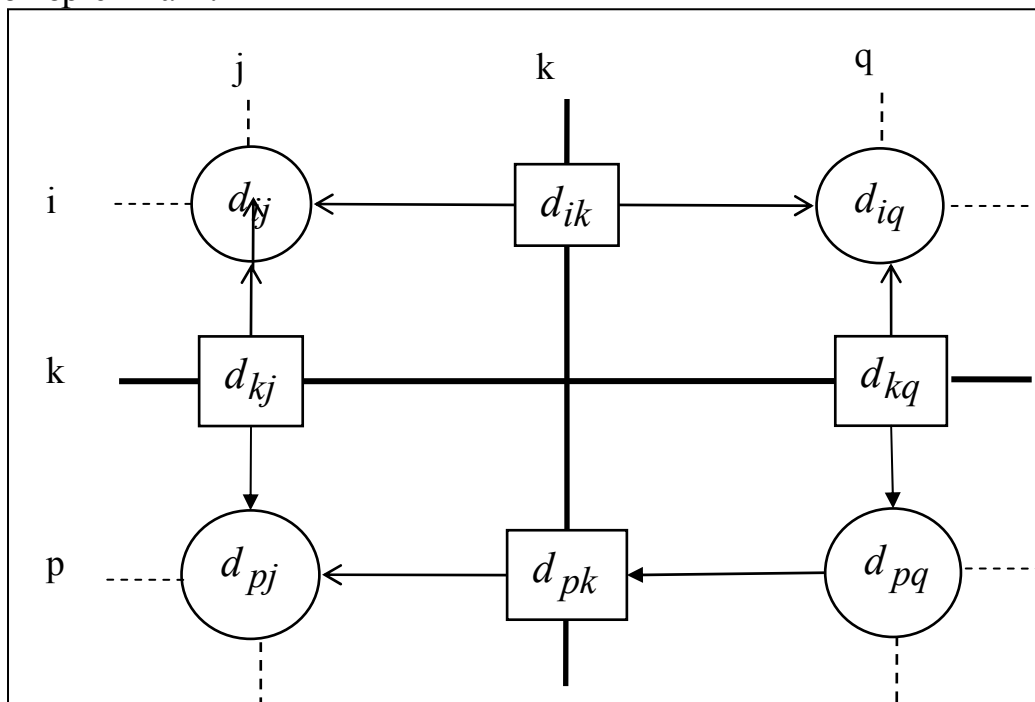


Рис. 5.2. Матрица  $D_{k-1}$  на  $k$  шаге алгоритма Флойда

Поясним алгоритм Флойда (рис. 5.2). Строка  $k$  и столбец  $k$  являются ведущими. Строка  $i$  – любая строка с номером от 1 до  $k-1$ , а строка  $p$  – произвольная строка с номером от  $k+1$  до  $n$ . Столбец  $j$  представляет любой столбец с номером от 1 до  $k-1$ , а столбец  $q$  – произвольный столбец с номером от  $k+1$  до  $n$ . Треугольный оператор выполняется следующим образом. Если сумма элементов ведущих строки и столбца (квадраты) меньше элементов, находящихся на пересечении столбца и строки (круги), соответствующих рассматриваемым ведущим элементам, то расстояние (элемент в круге) заменяется на сумму расстояний, представленных ведущими элементами.

После реализации  $n$  шагов алгоритма определение кратчайшего пути по матрицам  $D_n$  и  $S_n$ , кратчайшего пути между узлами  $i$  и  $j$  выполняется по правилам:

1. Расстояние между узлами  $i$  и  $j$  равно элементу  $d_{ij}$  в матрице  $D_n$ .
2. Промежуточные узлы пути от узла  $i$  к узлу  $j$  определяем по матрице  $S_n$ . Пусть  $s_{ij}=k$ , тогда имеем путь  $i \rightarrow k \rightarrow j$ . Если далее  $s_{ik}=k$  и  $s_{kj}=j$ , тогда считаем, что весь путь определен, так как найдены все промежуточные узлы. В противном случае повторяем описанную процедуру для путей от узла  $i$  к узлу  $k$  и от  $k$  к узлу  $j$ .

Пример нахождения кратчайшего пути по алгоритму Флойда.

Постановка задачи. На рис. 5.3 изображена сеть в виде графа. Расстояния между узлами сети проставлены возле соответствующих ребер. Ребро (3,5) ориентировано, поэтому не допускается движение от узла 5 к узлу 3. Все остальные ребра допускают движение в обе стороны. Найти кратчайшие пути между любыми двумя узлами.

Решение задачи.

1. Строятся начальные матрицы  $D_0$  (табл. 5.4) и  $S_0$  (табл. 5.5). Матрица  $D_0$  симметрична, за исключением пары элементов  $d_{35}$  и  $d_{53}$ , где  $d_{35} = \infty$ , поскольку невозможен переход от узла 5 к узлу 3.

2. В матрице  $D_0$  выделены ведущие строка и столбец ( $k=1$ ). Элементы  $d_{23}$  и  $d_{32}$  можно улучшить с помощью треугольного оператора. Получаем матрицы  $D_1$  (табл. 5.6) и  $S_1$  (табл. 5.7):

- Заменяем  $d_{23}$  на  $d_{21} + d_{13} = 3 + 10 = 13$  и устанавливаем  $s_{23} = 1$ .
- Заменяем  $d_{32}$  на  $d_{31} + d_{12} = 10 + 3 = 13$  и устанавливаем  $s_{32} = 1$ .



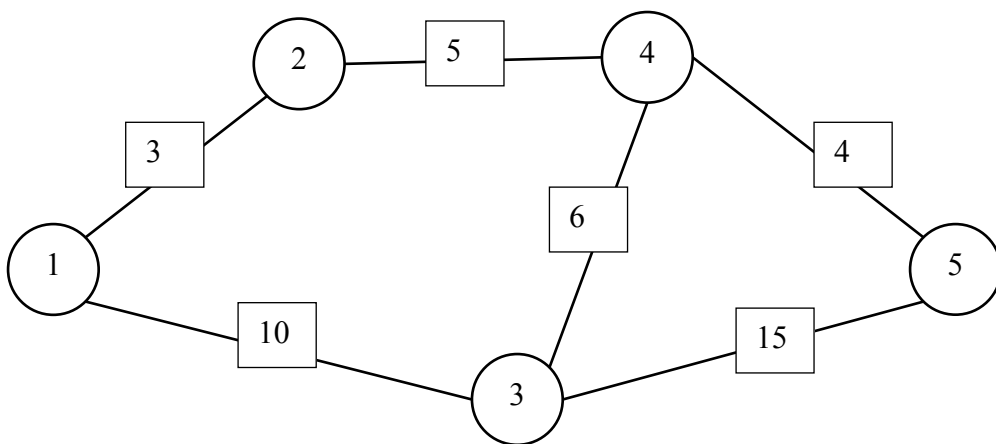


Рис. 5.3. Сеть, представленная в виде графа с весовыми коэффициентами

Таблица 5.4

Начальная матрица  $D_0$

Строки/Столбцы	1	2	3	4	5
1	-	3	10	$\infty$	$\infty$
2	3	-	$\infty$	5	$\infty$
3	10	$\infty$	-	6	15
4	$\infty$	5	6	-	4
5	$\infty$	$\infty$	$\infty$	4	-

3. Полагаем  $k=2$ . В матрице  $D_1$  выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матриц  $D_1$  и  $S_1$ , выделенных двойной рамкой. В результате получаем матрицы  $D_2$  (табл. 5.8) и  $S_2$  (табл. 5.9).

Таблица 5.5

Начальная матрица  $S_0$

Строки/Столбцы	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Таблица 5.6

Матрица  $D_1$ 

Строки/Столбцы	1	2	3	4	5
1	-	3	10	$\infty$	$\infty$
2	3	-	13	5	$\infty$
3	10	13	-	6	15
4	$\infty$	5	6	-	4
5	$\infty$	$\infty$	$\infty$	4	-

Таблица 5.7

Матрица  $S_1$ 

Строки/Столбцы	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

4. Полагаем  $k=3$ . В матрице  $D_2$  выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матриц  $D_2$  и  $S_2$ , выделенных двойной рамкой. В результате получаем матрицы  $D_3$  (табл. 5.10) и  $S_3$  (табл. 5.11).

Таблица 5.8

Матрица  $D_2$ 

Строки/Столбцы	1	2	3	4	5
1	-	3	10	8	$\infty$
2	3	-	13	5	$\infty$
3	10	13	-	6	15
4	$\infty$	5	6	-	4
5	$\infty$	$\infty$	$\infty$	4	-

Таблица 5.9

Матрица  $S_2$ 

Строки/Столбцы	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3		5
5	1	2	3	4	-

Таблица 5.10

Матрица  $D_3$ 

Строки/Столбцы	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	8	5	6	-	4
5	$\infty$	$\infty$	$\infty$	4	-

5. Полагаем  $k=4$ . В матрице  $D_3$  выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матриц  $D_3$  и  $S_3$ , выделенных двойной рамкой. В результате получаем матрицы  $D_4$  (табл. 5.12) и  $S_4$  (табл. 5.13).

Таблица 5.11

Матрица  $S_3$ 

Строки/Столбцы	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Таблица 5.12

Матрица  $D_4$ 

Строки/Столбцы	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

Таблица 5.13

Матрица  $S_4$ 

Строки/Столбцы	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

6. Полагаем  $k=5$ . В матрице  $D_4$  выделены ведущие строка и столбец. Никаких действий на шаге не выполняем. Вычисления закончены.

Конечные матрицы  $D_4$  и  $S_4$  содержат всю информацию, необходимую для определения кратчайших путей между любыми двумя узлами сети. Например, кратчайшее расстояние между узлами 1 и 5 равно  $d_{15}=12$ .

Сегмент маршрута  $(i,j)$  состоит из ребра  $(i,j)$ , если  $s_{ij}=j$ . Иначе узлы  $i$  и  $j$  связаны, по крайней мере, через один промежуточный узел. Например,  $s_{15}=4$  и  $s_{45}=5$ , сначала кратчайший маршрут между узлами 1 и 5 будет иметь вид  $1 \rightarrow 4 \rightarrow 5$ . Но так как  $s_{14}=4$ , узлы 1 и 4 в определяемом пути не связаны одним ребром (но в исходной сети они могут быть связаны непосредственно). Определяем промежуточный узел (узлы) между первым и четвертым узлами:  $s_{14}=2$  и  $s_{24}=4$ , поэтому маршрут  $1 \rightarrow 4$  заменяем  $1 \rightarrow 2 \rightarrow 4$ . Поскольку  $s_{12}=2$  и  $s_{24}=4$ , других промежуточных узлов нет. Комбинируя определенные сегменты маршрута, получаем кратчайший путь от узла 1 до узла 5:  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ . Длина пути равна 12 милям [13].

## **6. Задания для выполнения лабораторных работ**

### **6.1. Задания для выполнения лабораторной работы №1 «Линейные структуры данных»**

#### **Вариант 1**

Разработать программу, которая содержит динамическую информацию о наличии автобусов в автобусном парке.

Сведения о каждом автобусе содержат:

- номер автобуса;
- фамилию и инициалы водителя;
- номер маршрута.

Программа должна обеспечивать:

- начальное формирование данных в виде списка о всех автобусах;
- при выезде каждого автобуса из парка вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;

- при въезде каждого автобуса в парк вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;

- по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

#### **Вариант 2**

Разработать программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка содержит:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.

Программа должна обеспечивать:

- хранение всех заявок в виде списка;
- добавление заявок в список;
- удаление заявок;
- вывод заявок по заданному номеру рейса и дате вылета;
- вывод всех заявок.

### **Вариант 3**

Разработать программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

- номер УДК;
- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

- начальное формирование данных о всех книгах в библиотеке в виде списка;
- при взятии каждой книги вводится номер УДК, и программа уменьшает значение количества книг на единицу или выдает сообщение о том, что требуемой книги в библиотеке нет, или требуемая книга находится на руках;
- при возвращении каждой книги вводится номер УДК, и программа увеличивает значение количества книг на единицу;
- по запросу выдаются сведения о наличии книг в библиотеке.

### **Вариант 4**

Разработать программу, которая содержит динамическую информацию о наличии автобусов в автобусном парке.

Сведения о каждом автобусе содержат:

- номер автобуса;
- фамилию и инициалы водителя;
- номер маршрута;
- признак того, где находится автобус – на маршруте или в парке.

Программа должна обеспечивать:

- начальное формирование данных;
- при выезде каждого автобуса из парка вводится номер автобуса, и программа устанавливает значение признака «автобус на маршруте»;
- при въезде каждого автобуса в парк вводится номер автобуса, и программа устанавливает значение признака «автобус в парке»;
- по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

### **Вариант 5**

Разработать программу, отыскивающую проход по лабиринту.

Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице.

Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода выводит найденный путь в виде координат квадратов. Для хранения пути использовать стек.

### **Вариант 6**

Гаражная стоянка имеет одну стояночную полосу, причем въезд и выезд находятся в одном конце полосы. Если владелец автомашины приходит забрать свой автомобиль, который не является ближайшим к выходу, то все автомашины, загораживающие проезд, удаляются, машина данного владельца выводится со стоянки, а другие машины возвращаются на стоянку в исходном порядке.

Разработать программу, которая моделирует процесс прибытия и отъезда машин. Прибытие или отъезд автомашины задается командной строкой, которая содержит признак прибытия или отъезда и номер автомашины. Программа должна выводить сообщение при прибытии или выезде любой автомашины. При выезде автомашины со стоянки сообщение должно содержать число, которое показывает, сколько раз машина удалялась со стоянки для обеспечения выезда других автомобилей.

### **Вариант 7**

Разработать программу, моделирующую заполнение гибкого магнитного диска.

Общий объем памяти на диске 360 Кбайт. Файлы имеют произвольную длину от 18 до 32 Кбайт. В процессе работы файлы либо записываются на диск, либо удаляются с него.

В начале работы файлы записываются подряд друг за другом. После удаления файла на диске образуется свободный участок памяти, и вновь записываемый файл либо размещается на свободном участке, либо, если файл не вмещается в свободный участок, размещается после последнего записанного файла.

В случае, когда файл превосходит длину самого большого свободного участка, выдается аварийное сообщение. Требование на запись или удаление файла задается в командной строке, которая содержит имя файла, его длину в байтах, признак записи или удаления. Программа должна выдавать по запросу сведения о занятых и свободных участках памяти на диске.

Необходимо создать список занятых участков и список свободных участков на диске.

### **Вариант 8**

В файловой системе каталог файлов организован как линейный список.

Для каждого файла в каталоге содержатся следующие сведения:

- имя файла:

- дата создания;
- количество обращений к файлу;

Разработать программу, которая обеспечивает:

- начальное формирование каталога файлов;
- вывод каталога файлов;
- удаление файлов, дата создания которых меньше заданной;
- выборку файлов с наибольшим количеством обращений.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 9**

Предметный указатель организован как линейный список.

Каждая компонента указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до 10.

Разработать программу, которая обеспечивает:

- начальное формирование предметного указателя;
- вывод предметного указателя;
- вывод номера страниц для заданного слова.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 10**

Текст помощи для программы организован как линейный список.

Каждая компонента текста помощи содержит термин (слово) и текст, содержащий пояснения к этому термину. Количество строк текста, относящихся к одному термину, от одной до пяти.

Разработать программу, которая обеспечивает:

- начальное формирование текста помощи;
- вывод текста помощи;
- вывод поясняющего текста для заданного термина.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 11**

Картотека в бюро обмена квартир организована как линейный список.

Сведения о каждой квартире содержат:

- количество комнат;
- этаж;
- площадь;
- адрес.



Разработать программу, которая обеспечивает:

- начальное формирование картотеки;
- ввод заявки на обмен;
- поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 105 выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в список;
- вывод всего списка.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## **Вариант 12**

Анкета для опроса населения содержит две группы вопросов.

Первая группа содержит сведения о респонденте:

- возраст;
- пол;
- образование (начальное, среднее, высшее).

Вторая группа содержит собственно вопрос анкеты, ответ на который ДА или НЕТ.

Разработать программу, которая:

- обеспечивает начальный ввод анкет и формирует из них линейный список;
- на основе анализа анкет выдает ответы на следующие вопросы:
  - 1) сколько мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос анкеты;
  - 2) сколько женщин моложе 30 лет, имеющих среднее образование, ответили НЕТ на вопрос анкеты;
  - 3) сколько мужчин моложе 25 лет, имеющих начальное образование, ответили ДА на вопрос анкеты;
  - 4) производит вывод всех анкет и ответов на вопросы.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## **Вариант 13**

Разработать программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

- номер УДК;
- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

- начальное формирование данных о всех книгах в библиотеке в виде списка;
- добавление данных о книгах, вновь поступающих в библиотеку;
- удаление данных о списываемых книгах;
- по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.

#### **Вариант 14**

На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована как линейный список.

Разработать программу, которая:

- обеспечивает начальное формирование картотеки в виде линейного списка;
- производит вывод всей картотеки;
- вводит номер телефона и время разговора;
- выводит извещение на оплату телефонного разговора.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

#### **Вариант 15**

Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:

- номер поезда;
- станция назначения;
- время отправления;

Данные в информационной системе организованы в виде линейного списка.

Разработать программу, которая:

- обеспечивает первоначальный ввод данных в информационную систему и формирование линейного списка;
- производит вывод всего списка;
- вводит номер поезда и выводит все данные об этом поезде;
- вводит название станции назначения и выводит данные обо всех поездах, следующих до этой станции.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## **6.2 Задания для выполнения лабораторной работы №2 «Нелинейные структуры данных. Бинарные деревья»**

### **Вариант 1**

Разработать программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

- номер УДК;
- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

- начальное формирование данных о всех книгах в библиотеке в виде двоичного дерева;
- добавление данных о книгах, вновь поступающих в библиотеку;
- удаление данных о списываемых книгах;

По запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.

### **Вариант 2**

Разработать программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка содержит:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.

Программа должна обеспечивать:

- хранение всех заявок в виде двоичного дерева;
- добавление и удаление заявок;
- по заданному номеру рейса и дате вылета вывод заявок с их последующим удалением;
- вывод всех заявок.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 3**

Англо-русский словарь построен как двоичное дерево.

Каждая компонента содержит английское слово, соответствующее ему русское слово и счетчик количества обращений к данной компоненте.

Первоначально дерево формируется согласно английскому алфавиту. В процессе эксплуатации словаря при каждом обращении к компоненте в счетчик обращений добавляется единица.

Разработать программу, которая:

- обеспечивает начальный ввод словаря с конкретными значениями счетчиков обращений;

- формирует новое представление словаря в виде двоичного дерева по следующему алгоритму:

- 1) в старом словаре ищется компонента с наибольшим значением счетчика обращений;

- 2) найденная компонента заносится в новый словарь и удаляется из старого;

- 3) переход к этапу 1 до исчерпания исходного словаря;

- обеспечивает вывод исходного и нового словарей.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

#### **Вариант 4**

На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована как двоичное дерево.

Разработать программу, которая:

- обеспечивает начальное формирование картотеки в виде двоичного дерева;

- производит вывод всей картотеки;

- вводит номер телефона и время разговора;

- выводит извещение на оплату телефонного разговора.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

#### **Вариант 5**

Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования.

Для каждого поезда указывается:

- номер поезда;

- станция назначения;

- время отправления.

Данные в информационной системе организованы в виде двоичного дерева.

Разработать программу, которая:

- обеспечивает первоначальный ввод данных в информационную систему и формирование двоичного дерева;

- производит вывод всего дерева;

- вводит номер поезда и выводит все данные об этом поезде;

- вводит название станции назначения и выводит данные о всех поездах, следующих до этой станции.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 6**

Разработать программу, которая содержит информацию об автомобилях в автосалоне.

Сведения об автомобилях содержат:

- модель автомобиля;
- основные характеристики (тип кузова, количество дверей/мест, длина/ширина/высота, колесная база (мм), снаряженная масса автомобиля (кг), полная масса автомобиля (кг));
- характеристики двигателя (тип двигателя, рабочий объем, степень сжатия, максимальная мощность (л.с.), максимальный крутящий момент (Нм/об.мин));
- скоростные характеристики (максимальная скорость (км/ч), средний расход топлива (л/100 км), тип потребляемого бензина);
- количество автомобилей данной модели в автосалоне.

Программа должна обеспечивать:

- начальное формирование данных обо всех автомобилях в автосалоне в виде двоичного дерева;
- добавление данных об автомобилях, вновь поступающих в автосалон;
- удаление данных о проданных автомобилях.

По запросу выдаются сведения о наличии автомобиля в автосалоне, упорядоченные по наименованию модели автомобилей.

### **Вариант 7**

Разработать программу, которая содержит текущую информацию о заявлениях на приобретение туристических путевок.

Каждое заявление содержит:

- название страны посещения;
- время года;
- фамилию и инициалы отдыхающего;
- желаемую дату отправления;
- сроки пребывания;
- уровень обслуживания.

Программа должна обеспечивать:

- хранение всех заявлений в виде двоичного дерева;
- добавление и удаление заявлений;
- вывод заявлений по фамилии и инициалам отдыхающих с их последующим удалением;
- вывод всех заявок.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 8**

Разработать программу, которая содержит информацию об абонентах кабельного телевидения.

Сведения об абонентах содержат:

- фамилию и инициалы;
- адрес;
- телефон;
- номер договора.

Программа должна обеспечивать:

- начальное формирование данных обо всех абонентах кабельного телевидения в виде двоичного дерева;
- добавление данных об абонентах, вновь поступающих в картотеку;
- удаление данных об абонентах из картотеки, разорвавших договор;
- по запросу выдавать сведения о наличии абонентов в картотеке, упорядоченные по номеру договора.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 9**

Разработать программу, которая содержит информацию о безработных, зарегистрированных на бирже труда.

Данные о зарегистрированных безработных содержат:

- номер регистрации безработного;
- фамилия, имя, отчество;
- возраст;
- пол;
- образование;
- профессия;
- общий стаж работы;
- дата постановки на учет;
- желаемая заработная плата;
- желаемая должность.

Программа должна обеспечивать:

- хранение всех зарегистрированных безработных в виде двоичного дерева;
- добавление данных о безработных;
- удаление данных о безработных, нашедших работу;
- вывод данных о безработных по фамилии, имени, отчеству, регистрационному номеру;
- вывод всех зарегистрированных безработных.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 10**

Разработать программу, которая содержит информацию о дилерах компании.

Сведения о дилерах содержат:

- адрес;
- фамилия, имя, отчество;
- телефон;
- электронный адрес;
- объем закупок продукции в месяц;
- объем продаж продукции за месяц;
- льготный процент скидки при закупке продукции;

Программа должна обеспечивать:

- начальное формирование данных обо всех дилерах фирмы в виде двоичного дерева;
- добавление данных о дилерах;
- удаление данных о дилерах;
- выдавать сведения о дилерах по фамилии, имени, отчеству.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### **Вариант 11**

Разработать программу, которая содержит информацию о дисциплинах, читаемых преподавателем студентам университета в течение учебного года.

Сведения о нагрузке преподавателя за учебный год содержат:

- название дисциплины;
- семестр проведения занятия;
- количество студентов;
- количество часов аудиторных лекций;
- количество часов аудиторных практических работ;
- вид контроля знаний студентов (зачет/экзамен).

Программа должна обеспечивать:

- начальное формирование данных о дисциплинах, читаемых преподавателем, в виде двоичного дерева;
- добавление данных о дисциплинах;
- удаление данных о дисциплинах;
- вывод данных о дисциплинах по наименованию;
- вывод всех дисциплин, составляющих нагрузку преподавателя.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## Вариант 12

Разработать программу, которая содержит информацию о реестре жилых помещений (купля/продажа) риэлторской фирмы.

Данные реестра жилья содержат:

- район;
- адрес;
- количество комнат;
- общая площадь;
- жилая площадь;
- год постройки дома;
- стоимость.

Программа должна обеспечивать:

- хранение всех данных о жилых помещениях в виде двоичного дерева;
- добавление в реестр данных о жилых помещениях;
- удаление данных о проданных жилых помещениях из реестра фирмы;
- вывод данных о жилых помещениях по стоимости;
- вывод всех жилых помещений, занесенных в реестр риэлторской фирмы.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## Вариант 13

Разработать программу, которая содержит информацию о моделях компьютеров, продаваемых в магазине вычислительной техники.

Сведения о компьютере содержат:

- марка компьютера;
- тип процессора;
- тактовая частота процессора;
- объем памяти;
- объем жесткого диска;
- объем памяти видеокарты;
- цена компьютера;
- количество экземпляров, имеющих в наличии.

Программа должна обеспечивать:

- начальное формирование данных обо всех компьютерах в магазине вычислительной техники в виде двоичного дерева;
- добавление данных о компьютерах, поступающих в магазин;
- удаление данных о проданных компьютерах;
- выдавать сведения о наличии компьютеров в магазине, упорядоченные по наименованию модели.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.



## Вариант 14

Разработать программу, которая содержит информацию о сотрудниках, работающих в фирме.

Сведения о сотрудниках содержат:

- табельный номер;
- фамилия, имя, отчество;
- образование;
- год поступления на работу;
- домашний адрес;
- оклад.

Программа должна обеспечивать:

- начальное формирование данных обо всех сотрудниках фирмы в виде двоичного дерева;
- добавление данных о сотрудниках, вновь принятых на работу;
- удаление данных о сотрудниках, уволенных с работы;
- по запросу выдавать сведения о сотрудниках в штате фирмы, упорядоченные по фамилии, имени, отчеству.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

## Вариант 15

Автоматизированная информационная система на автовокзале содержит сведения об отправлении пригородных автобусов.

Для каждого автобуса указывается:

- номер автобуса;
- пункт назначения;
- время отправления,
- время прибытия.

Данные в информационной системе организованы в виде двоичного дерева.

Разработать программу, которая:

- обеспечивает первоначальный ввод данных в информационную систему и формирование двоичного дерева;
- производит вывод всего дерева;
- вводит номер автобуса и выводит все данные об автобусе;
- вводит название пункта назначения и выводит данные о всех автобусах, следующих до этого населенного пункта.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

### 6.3. Задания для выполнения лабораторной работы №3 «Алгоритмы сортировки данных в оперативной памяти»

#### Вариант 1

Разработать программу, которая выполняет:

1. Генерацию массива числовых данных размера  $N$  со случайным распределением значений элементов массива, подчиняющихся вероятностному закону распределения.

2. Сортировку исходного массива простыми и сложными методами (по 3 каждого вида).

3. Провести сравнительный анализ простых и сложных методов сортировки элементов массива, рассчитав показатели производительности (время сортировки и соотношение методов производительности (относительное время сортировки)). Результаты анализа представить в табличной форме записи.

4. Определить оценку качества, реализованных в программе простых методов сортировки (выбором, вставками, обменом) по двум показателям:

$$\frac{C}{N} \text{ и } \frac{M}{N},$$

где  $C$  – количество операций сравнения элементов массива;

$M$  – количество перестановок элементов массива, потребовавшихся для сортировки массива.

Результаты представить в табличной форме записи.

Количество элементов в массиве  $N$  и вероятностный закон распределения по вариантам приведены в табл. 6.1

*Таблица 6.1*

*Исходные данные*

Вариант	Количество элементов в массиве $N$	Вероятностный закон распределения
1	9000	Bradford
2	3000	Burr
3	5000	Cauchy
4	8000	Exponential
5	6000	ExtremeLB
6	2000	Fisk
7	5000	Gumbel
8	6000	Laplace
9	4000	Logistic
10	6000	Pareto

11	7000	Reciprocal
12	10000	Weibull
13	2000	Равномерное
14	3000	Эрланга
15	4000	Гиперэкспоненциальное

#### 6.4. Задания для выполнения лабораторной работы №4 «Внутренний поиск данных в таблице»

##### Вариант 1

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности ООО «Центр оценки и продажи недвижимости». Одним из источников прибыли этой организации является покупка и продажа квартир. Центр оценки имеет большой штат специалистов, позволяющий этой организации проводить сделки купли-продажи на высоком профессиональном уровне. Владелец квартиры, желающий ее продать, заключает договор с Центром, в котором указывается сумма, срок продажи и процент отчислений в пользу Центра оценки и продажи недвижимости в случае успешного проведения сделки. Один клиент может заключить с Центром более одного договора купли-продажи одновременно, если он владеет несколькими квартирами. Обмен квартир специалисты центра непосредственно не производят. Для этих целей используется вариант купли-продажи.

##### Вариант 2

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности отдела вневедомственной охраны квартир. Этот отдел обеспечивает электронную охрану квартир граждан в одном районе города. Для установки охранной сигнализации требуется наличие квартирного телефона. Один гражданин может заключить договор на охрану нескольких квартир. Из-за ложных срабатываний сигнализации возможно несколько выездов патрульных экипажей по одной квартире. На владельца квартиры, вовремя не отключившего сигнализацию после своего прихода домой, налагается штраф, величина которого оговаривается при заключении договора охраны. Если отдел вневедомственной охраны не уберег имущество владельца квартиры, то он выплачивает пострадавшему заранее оговоренную сумму. От величины этой суммы зависит размер ежемесячной оплаты за охрану квартиры.

##### Вариант 3

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности отдела приватизации жилья администрации города. В нашем городе на начало 2001 г. приватизировано около 80 000 квартир

граждан. Еще далеко не все проживающие в «своих» квартирах стали собственниками своего жилья. Процесс приватизации продолжается и займет еще несколько лет. Главная задача программного комплекса – не допустить приватизации одним человеком более одной квартиры. К сожалению, в отделе приватизации не используется уникальный кадастровый номер здания, поэтому вам придется использовать составной первичный ключ (адрес) для таблицы зданий, квартир и проживающих. Помните, что некоторые из проживающих в квартире могут не участвовать в приватизации.

#### **Вариант 4**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности предприятия «Газкомплект» по учету платы за пользование газом и газовыми приборами. Плата взимается с каждой квартиры в зависимости от количества потребленного газа или от числа проживающих, если счетчик отсутствует. Ответственный квартиросъемщик обязан каждый месяц снимать показания счетчика и производить оплату за потребленный газ через Сбербанк. Наряду с отслеживанием платы за газ предприятие производит профилактическое обслуживание газовых приборов. Правила техники безопасности предусматривают осмотр газовой плиты инспектором предприятия раз в квартал. Если обнаружены неполадки в подключении плиты или ее работе, то работник предприятия обязан немедленно устранить их за счет абонента. Оплата оказанных услуг осуществляется на месте по квитанции.

#### **Вариант 5**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности «Бюро технической инвентаризации» по изготовлению и выдаче технических паспортов на объекты недвижимости. Перед регистрацией сделки с объектом недвижимости собственник объекта должен получить на него технический паспорт в БТИ. Ежедневно в БТИ обращается до 200 человек. Основное назначение программного комплекса – не пропустить ни одного документа. Если технический паспорт не готов в назначенный срок, то БТИ должно выплатить неустойку. Алгоритм изготовления документа следующий. Клиент обращается к инспектору, сдает ему необходимые справки, согласовывает дату выхода техника на обмер, уплачивает аванс. Инспектор передает заявку начальнику отдела. Начальник отдела назначает исполнителя и техника. Техник выполняет обмер объекта. Исполнитель изготавливает документ и передает в отдел выдачи. В назначенный срок клиент забирает готовый документ, доплатив недостающую сумму. Один клиент (физическое или юридическое лицо) может заказать несколько технических паспортов, за изготовление которых оплата может производиться частями.

## **Вариант 6**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности отдела аренды ЗАО «Сириус». После удачной приватизации, когда у руководства этого предприятия оказалась большая часть акций, дела некогда мощного предприятия пошли на спад. Основная часть работников была уволена по сокращению штатов. В настоящее время основной статьей получения прибыли является сдача в аренду другим предприятиям и организациям площадей, которыми владеет «Сириус». В его собственности имеется 12-этажное здание, которое состоит примерно из 300 помещений. Почти все они сдаются в аренду. Один арендатор может арендовать несколько помещений, причем срок аренды для каждого устанавливается отдельно. Величина арендной платы и ее периодичность устанавливается арендодателем. После окончания срока аренды договор может быть продлен на прежних или новых условиях. Субаренда площадей запрещена. Закрытые договоры не удаляются из базы данных для отслеживания предыдущих арендаторов.

## **Вариант 7**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности телефонной компании. Основное назначение программного комплекса – отслеживание абонентской платы за телефоны. Клиентами компании могут быть как физические лица, так и организации. Расчет с организациями ведется в безналичной форме через банк. Физические лица вносят плату через кассу компании. Клиент телефонной компании может иметь несколько телефонных номеров. Дополнительная плата за подключенный параллельно аппарат не взимается. Если телефон у абонента не работает более суток, то плата за пользование телефоном уменьшается. Междугородние и международные звонки оплачиваются отдельно по заранее установленным расценкам.

## **Вариант 8**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности мелкооптового книжного магазина. Менеджер магазина, изучив спрос на книжную продукцию в городе, принимает решение о закупке партии книг в том или ином издательстве. Некоторые пользующиеся повышенным спросом книги могут быть закуплены у посредников. Часть продукции заказывается через Internet. Покупателем в мелкооптовом магазине может быть любой человек или организация, при условии, что величина покупки превысит одну тысячу рублей. Расчет с организациями производится через банк. Расчет с физическими лицами – наличными. Покупателю выписывается счет-фактура, которая имеет уникальный номер и содержит список книг с указанием их стоимости. После уплаты указанной суммы покупатель получает товар на складе.

## **Вариант 9**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности ОАО «Автовокзал». Это открытое акционерное общество занимается междугородними пассажирскими перевозками по Дальневосточному региону. В его собственности находится несколько десятков автобусов различной вместимости. Штат водителей превышает количество автобусов. Средний уровень сменности для машины – 2.5. Водитель не может работать более одной смены в сутки. Билеты на рейсы продаются только в здании автовокзала. Возможна предварительная продажа. Маршрут автобуса может пролегать через несколько населенных пунктов. В этом случае пассажир может купить билет до любого промежуточного пункта. Освободившимся местом после выхода пассажира распоряжается водитель. Полученную выручку он сдает в кассу предприятия после прибытия с маршрута. На линии работает контроль. Если в автобусе будет обнаружен пассажир без билета, то на водителя налагается штраф.

## **Вариант 10**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности агентства знакомств. Агентство занимается организацией знакомств одиноких мужчин и женщин. Возможен один из двух вариантов: человек либо регистрируется в агентстве, оставляет информацию о себе, чтобы любой мог ознакомиться с его анкетой, либо знакомится с базой уже зарегистрированных клиентов и выбирает подходящий вариант. Регистрация и подбор кандидатуры – платные. Тот, кто делает выбор по базе, платит за каждый выбранный вариант. После того, как выбор сделан, агентство согласовывает дату и время встречи с каждой из сторон, формирует и передает приглашения для знакомства обеим сторонам. Во избежание недоразумений первая встреча происходит в агентстве. Клиенты, желающие быть исключенными из базы, переносятся в архив.

## **Вариант 11**

Разработать консольное приложение, осуществляющее поиск в таблице данных деятельности ломбарда. Человек обращается в ломбард в том случае, если ему срочно нужны деньги. Например, недостает небольшой суммы для покупки квартиры, а подходящая квартира как раз продается, и на неё уже есть и другие покупатели. Тогда человек может пойти в ломбард и заложить вещи на необходимую сумму. В ломбарде с клиентом заключается договор. В нем оговариваются следующие условия: до какого срока выкуп вещи возможен без процентов, с какого времени будет взиматься процент, по истечении какого срока выкуп вещи невозможен, и она поступает в собственность ломбарда. Невыкупленные вещи ломбард выставляет на продажу.

## **Вариант 12**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности гостиницы. В гостинице существует большое количество возможных вариантов заселения гостей: все номера различаются по категориям (суперлюкс, люкс и т. д.), по количеству комнат в номере, количеству мест в каждом номере, а также по обустройству комнат – учитывается, например, наличие телевизора, холодильника, телефона. В обязанности администратора гостиницы входит подбор наиболее подходящего для гостя варианта проживания, регистрация гостей, прием платы за проживание, оформление квитанций, выписка отъезжающих. Учитывается также возможность отъезда гостя раньше указанного при регистрации срока, при этом производится перерасчет. Существует также услуга бронирования номера.

## **Вариант 13**

Разработать консольное приложение, осуществляющее поиск в таблице данных института селекции растений. Данный институт занимается сбором, выведением и продажей различных сортов семян. В его ассортименте можно найти семена практически всех возможных видов растений: от помидоров до редких цветов. Только что выведенные сорта заносятся в отдельный список для дальнейшего тестирования. Каждый сорт семян имеет свои характеристики, такие как урожайность, морозоустойчивость, адаптация к местным условиям, сроки созревания (раннеспелый, среднеспелый, поздний) и т. п. Покупатель может выбрать сорт, отвечающий тем или иным характеристикам. Компания занимается как оптовыми, так и розничными продажами. Оптовые покупатели заносятся в базу главным образом для того, чтобы информировать их о поступлении новых или отсутствовавших в определенный момент в продаже сортов.

## **Вариант 14**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности приемной комиссии университета. Каждый год университет зачисляет новых абитуриентов для возможного их поступления в университет после сдачи вступительных экзаменов. На бюджетную основу могут быть зачислены: абитуриенты, получившие на школьном экзамене высокий балл ЕГЭ и успешно прошедшие собеседование; абитуриенты, набравшие необходимый для бесплатного поступления балл на университетских экзаменах, а также абитуриенты, имеющие направление от какого-либо государственного предприятия. Все остальные могут поступить в университет на платной основе, набрав необходимое установленное университетом число баллов на вступительных экзаменах.

## **Вариант 15**

Разработать консольное приложение, осуществляющее поиск в таблице данных о деятельности кассы авиакомпании. Касса авиакомпании занимается продажей билетов на предстоящие рейсы. В билете указывается номер и название рейса, а также все остальные необходимые для пассажира данные: дата и время вылета, прибытия, номер места и класс (бизнес, экономический). Цена билета зависит от рейса, лайнера, класса, а также от времени покупки билета – иногда авиакомпании делают скидки купившим билет более чем за месяц или на «горящие рейсы» – все зависит от желания компании. Билеты продаются только совершеннолетним гражданам при предъявлении паспорта. У авиакомпании обычно имеется несколько касс, расположенных в разных концах города, поэтому обязательно необходимо учитывать номер кассы, в которой был продан билет, во избежание недоразумений при сдаче или обмене билета.

### **6.5. Задания для выполнения лабораторной работы №5 «Алгоритмы поиска на графе»**

#### ***6.5.1. Поиск кратчайшего пути. Алгоритм Дейкстры***

##### **Вариант 1**

Компания Bell Electric Company построила коммуникационную сеть между двумя приемно-передающими станциями 1 и 7. На рис. 6.1 коммуникационная сеть представлена в виде ориентированного графа. Вес дуги – расстояние в километрах. Определить кратчайшее расстояние между узлом 1 и всеми остальными узлами коммуникационной сети.

##### **Вариант 2**

Транспортная компания осуществляет перевозку автомобилей. На рис. 6.2 представлена транспортная сеть в виде ориентированного графа с 9 узлами. Вес дуги – расстояние в километрах. Определить кратчайшее расстояние между узлом 1 и всеми остальными узлами транспортной сети.



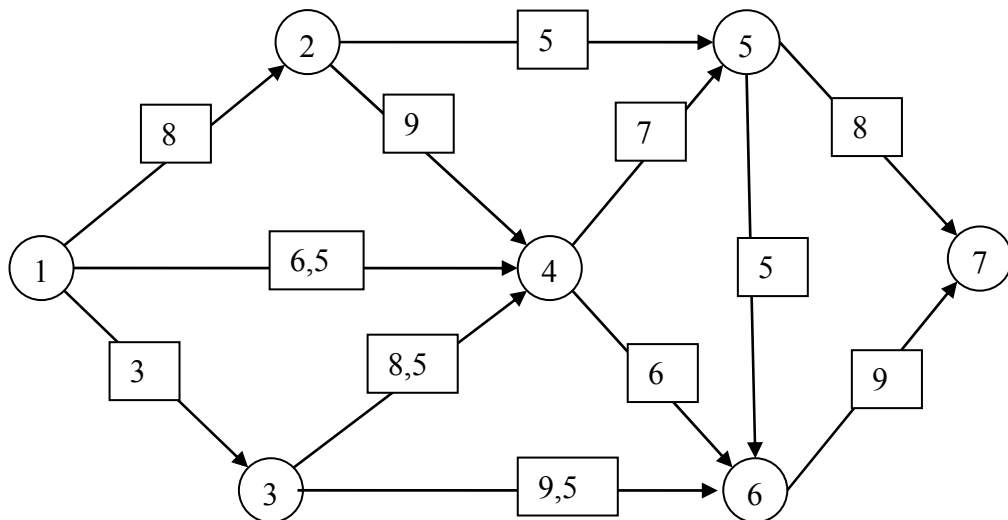


Рис. 6.1. Коммуникационная сеть компании Bell Electric Company

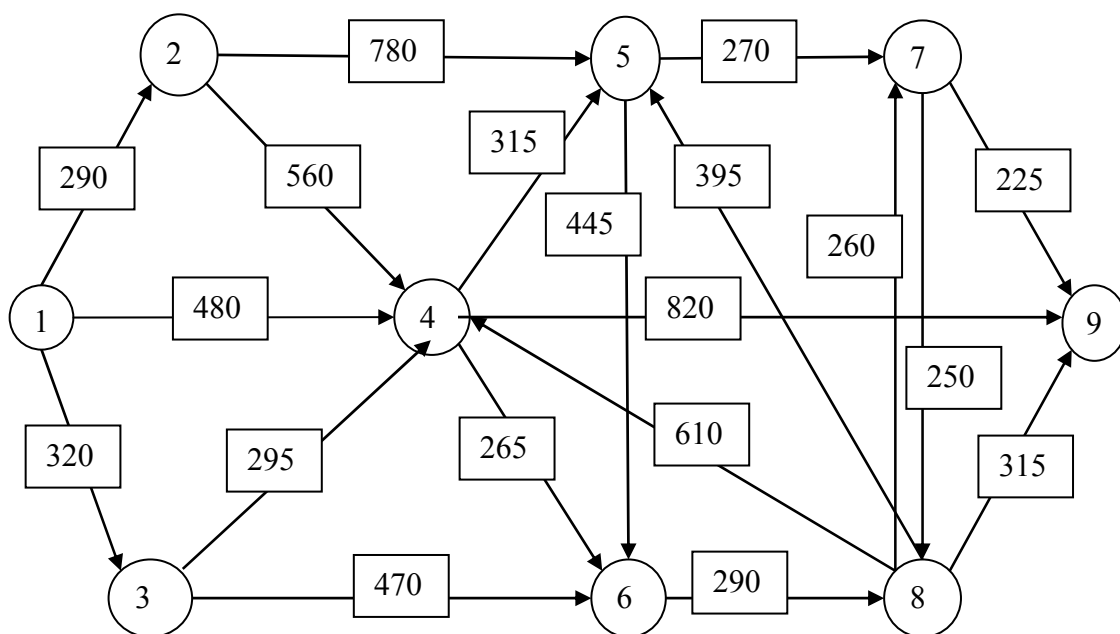


Рис. 6.2. Транспортная сеть перевозки автомобилей

### Вариант 3

Частное охранный предприятие осуществляет доставку денежных сумм из магазинов в банк. Транспортная сеть представлена в виде ориентированного графа с 10 узлами (рис. 6.3). Вес дуги – расстояние в километрах. Определить кратчайшие пути из узла 1 ко всем узлам транспортной сети.

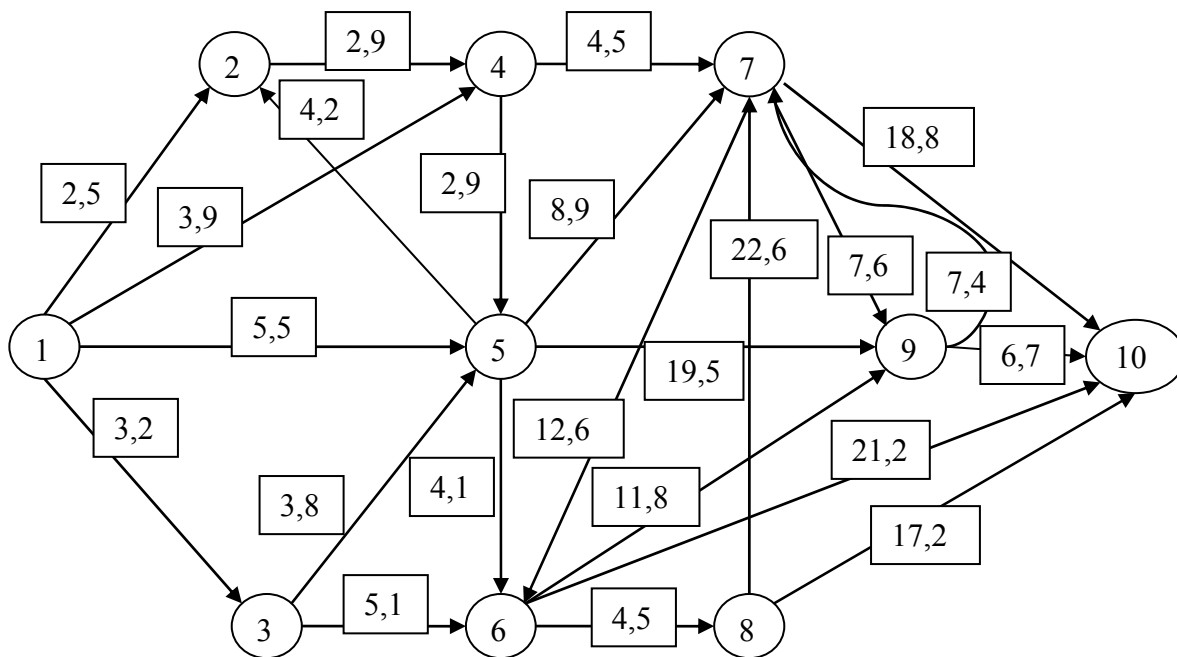


Рис. 6.3. Транспортная сеть перевозки денежных средств

#### Вариант 4

Торговая компания имеет филиалы в 8 точках города. Транспортная сеть с указанием расстояний в километрах представлена на рис. 6.4. Определить кратчайшие пути между узлом 1 и всеми остальными узлами транспортной сети.

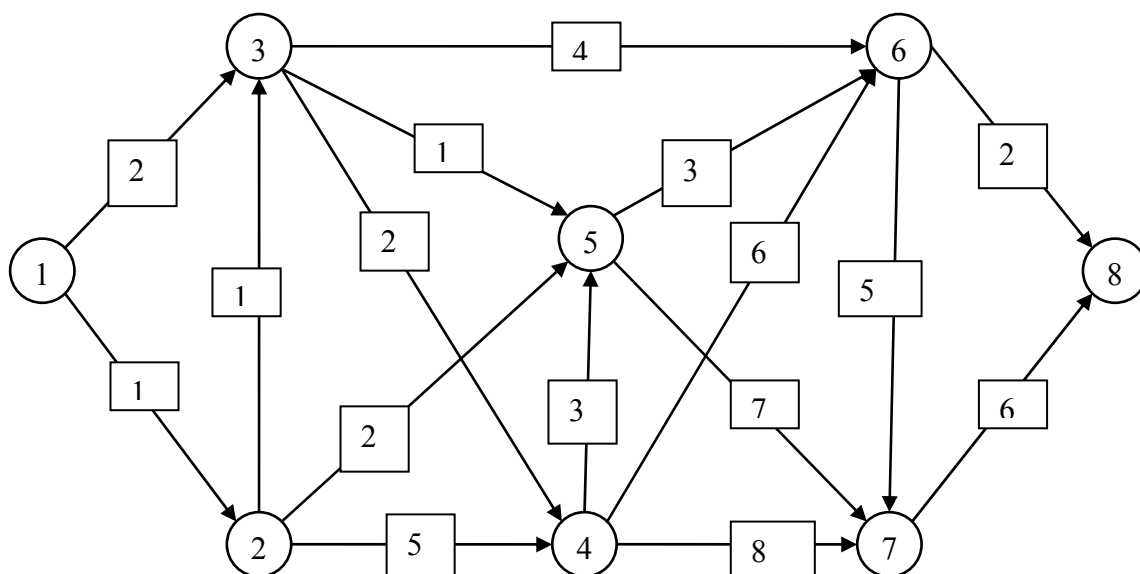


Рис. 6.4. Транспортная сеть торговой компании

### Вариант 5

Авиакомпания осуществляет перевозку грузов. На рис. 6.5 представлена транспортная сеть перевозок с указанием расстояний в тысячах километров. Определить кратчайшие расстояния между узлом 1 и всеми остальными узлами транспортной сети.

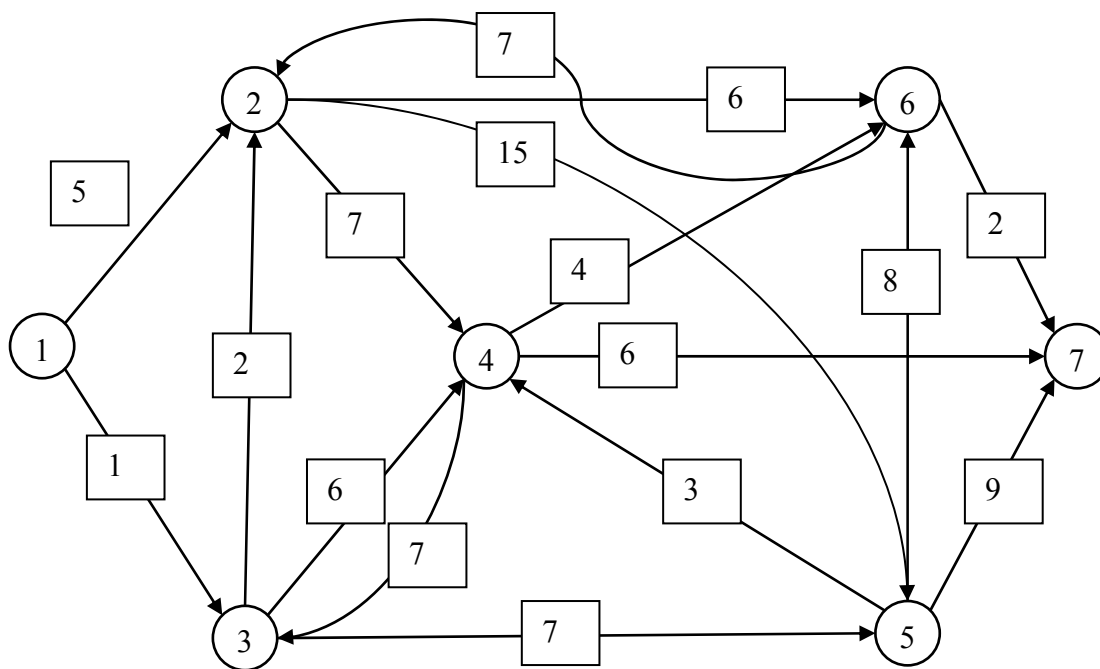


Рис. 6.5. Транспортная сеть авиакомпании

### Вариант 6

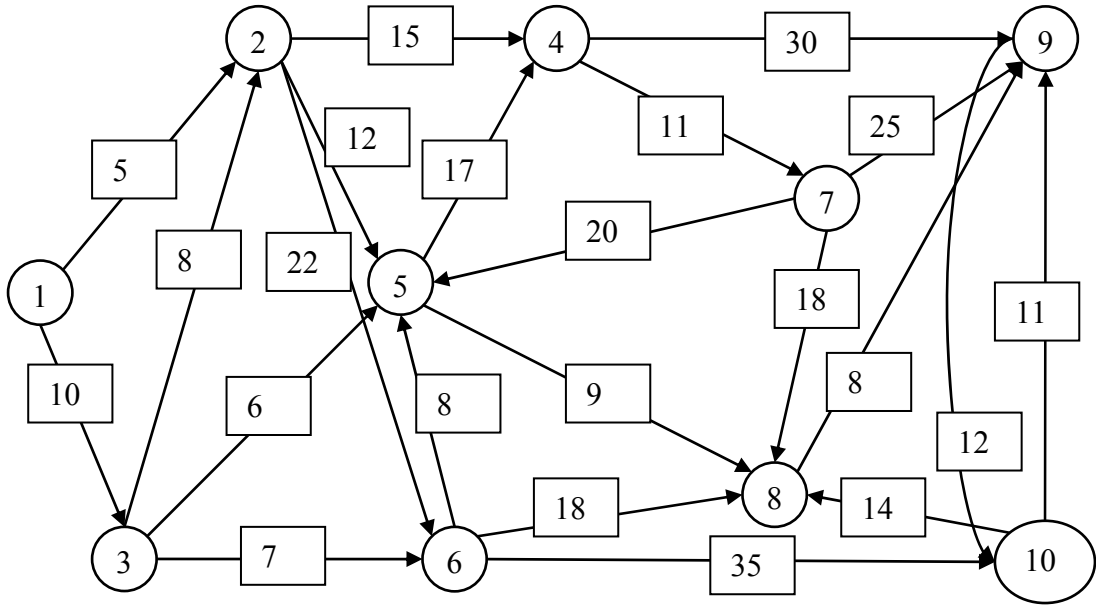
Компания по перевозке пассажиров осуществляет поездки в 10 населенных пунктов района. На рис. 6.6 представлены маршруты перевозки пассажиров с указанием расстояния в километрах. Определить кратчайшие пути между узлом 1 и всеми остальными узлами ориентированного графа.

### Вариант 7

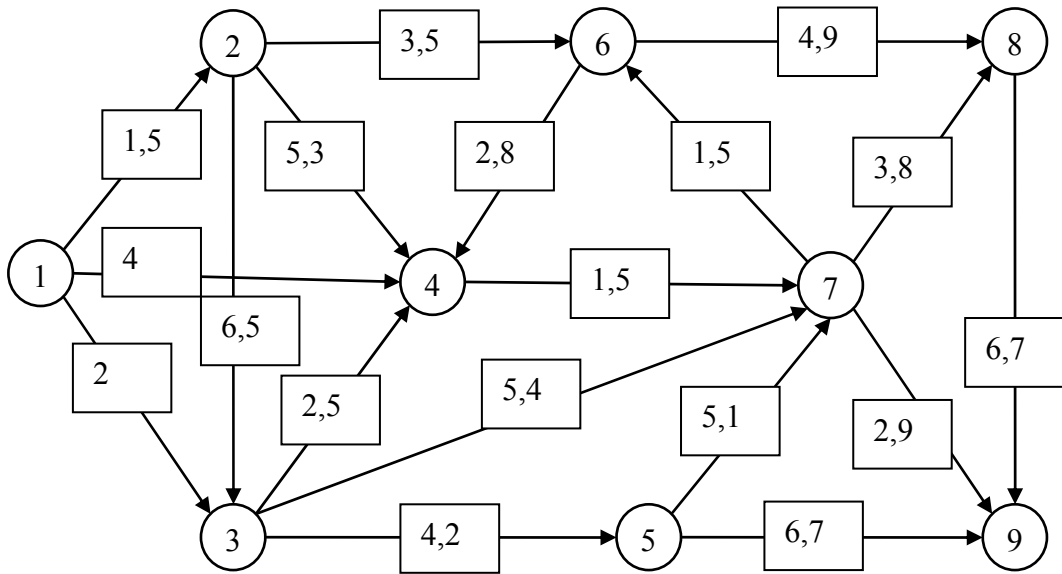
Железнодорожная компания выполняет перевозку грузов в 9 городов страны. Транспортная сеть маршрутов перевозок грузов представлена на рис. 6.7 в виде ориентированного графа. Расстояния указаны в тысячах километров. Определить кратчайшие пути между узлом 1 и всеми остальными узлами графа.

## Вариант 8

Торговая фирма имеет филиалы в 12 населённых пунктах области. Транспортная сеть показана на рис. 6.8 в виде ориентированного графа с указанием расстояния в километрах. Найти кратчайшие пути между узлом 1 и всеми остальными узлами графа.



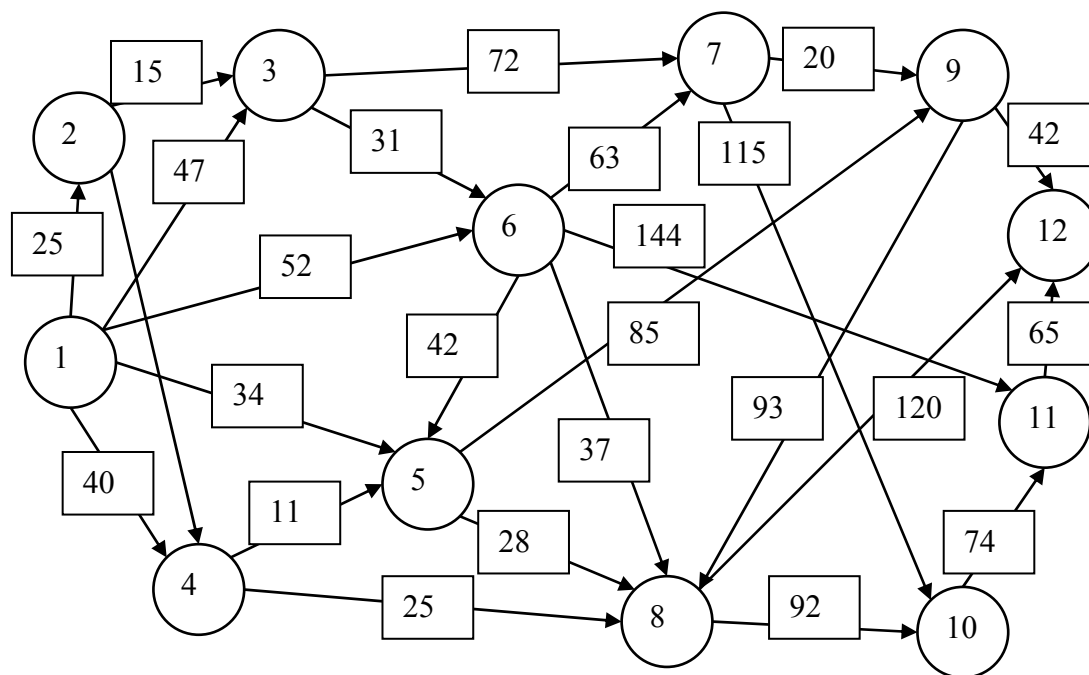
*Рис. 6.6. Маршруты перевозки пассажиров*



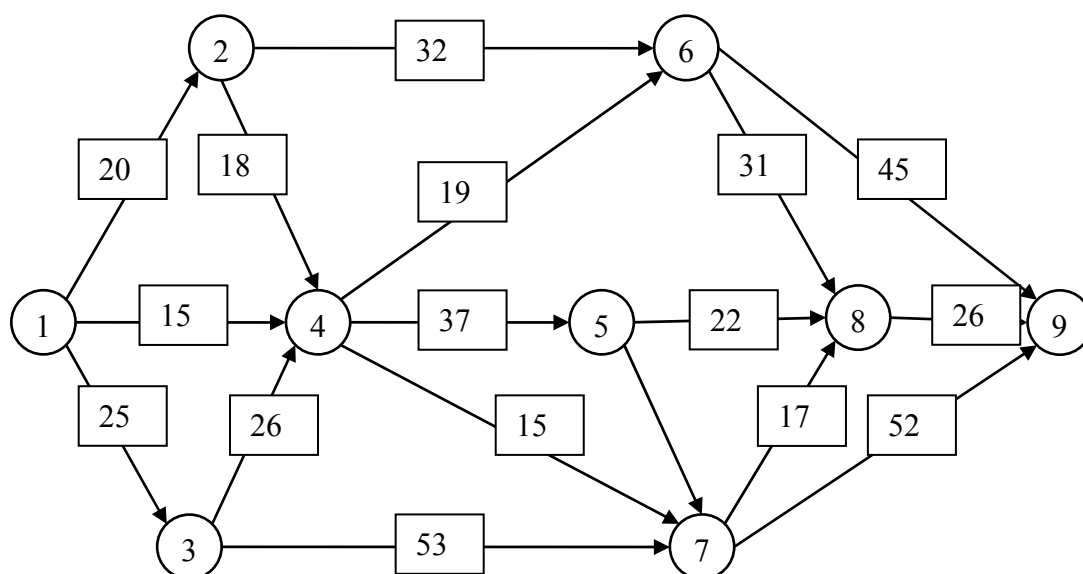
*Рис. 6.7. Транспортная сеть маршрутов перевозок железнодорожной компании*

## Вариант 9

Торговая компания имеет филиалы в 9 населённых пунктах. Сеть филиалов представлена в виде ориентированного графа (рис. 6.9). Вес дуги ориентированного графа - расстояние в километрах. Определить кратчайшие пути между узлом 1 и всеми остальными узлами графа.



*Рис. 6.8. Транспортная сеть торговой фирмы*



*Рис. 6.9. Транспортная сеть филиалов торговой фирмы*

### Вариант 10

Логистическая компания проектирует (нефтепровод) газопровод между 10 населенными пунктами. Транспортная сеть показана на рис. 6.10. Расстояния указаны в километрах. Определить кратчайший путь между узлами 1 и всеми остальными узлами.

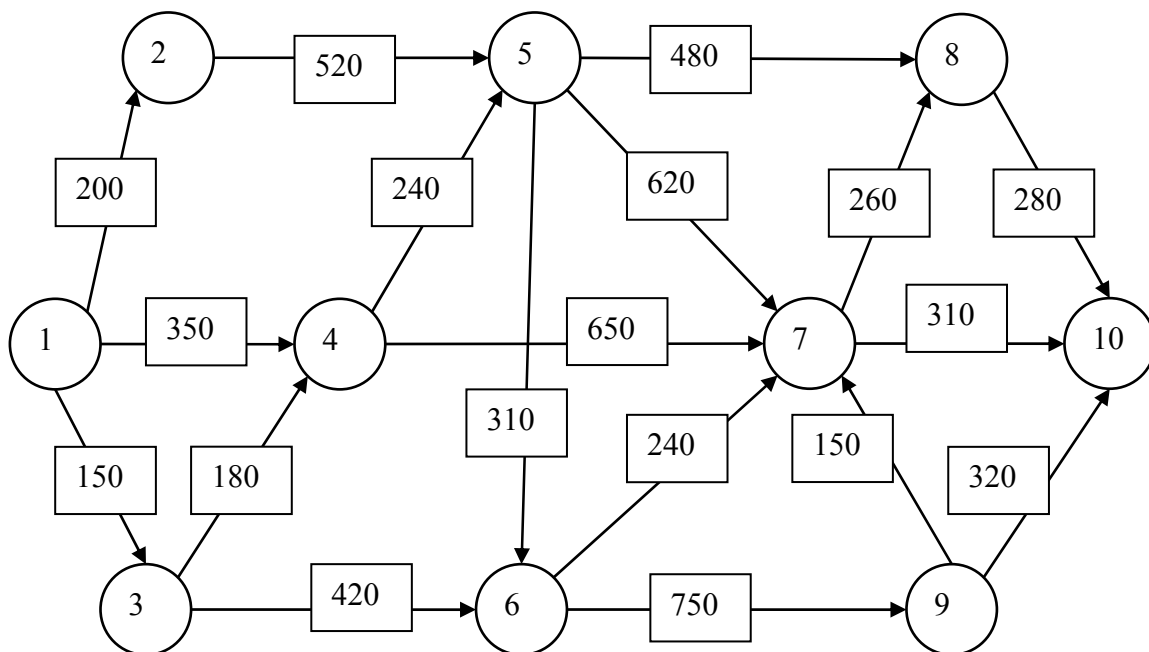
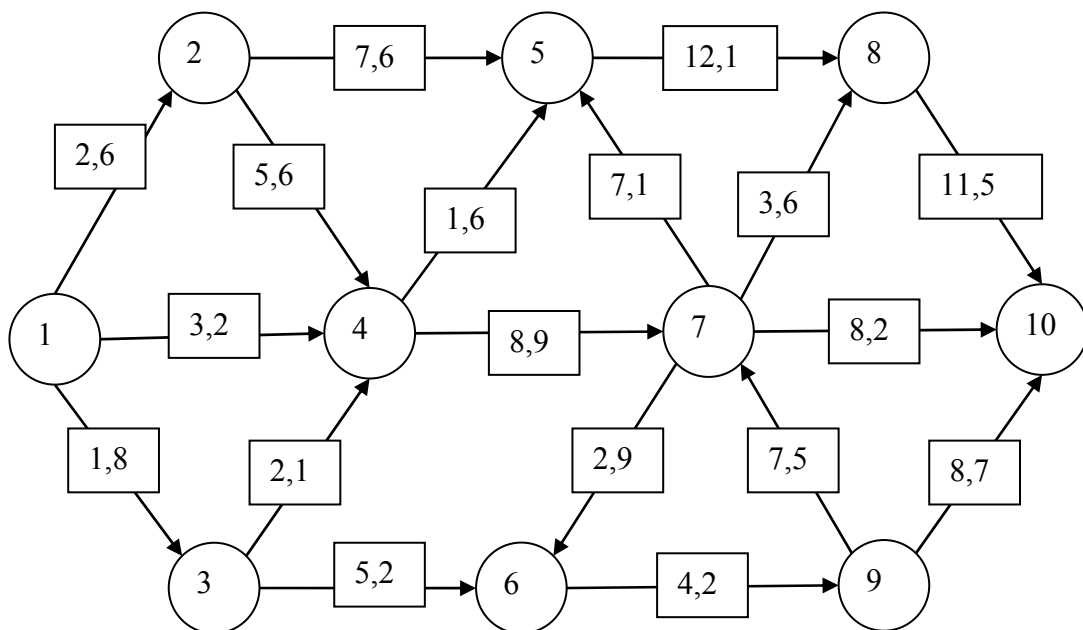


Рис. 6.10. Транспортная сеть газопровода, спроектированная логистической компанией

### Вариант 11

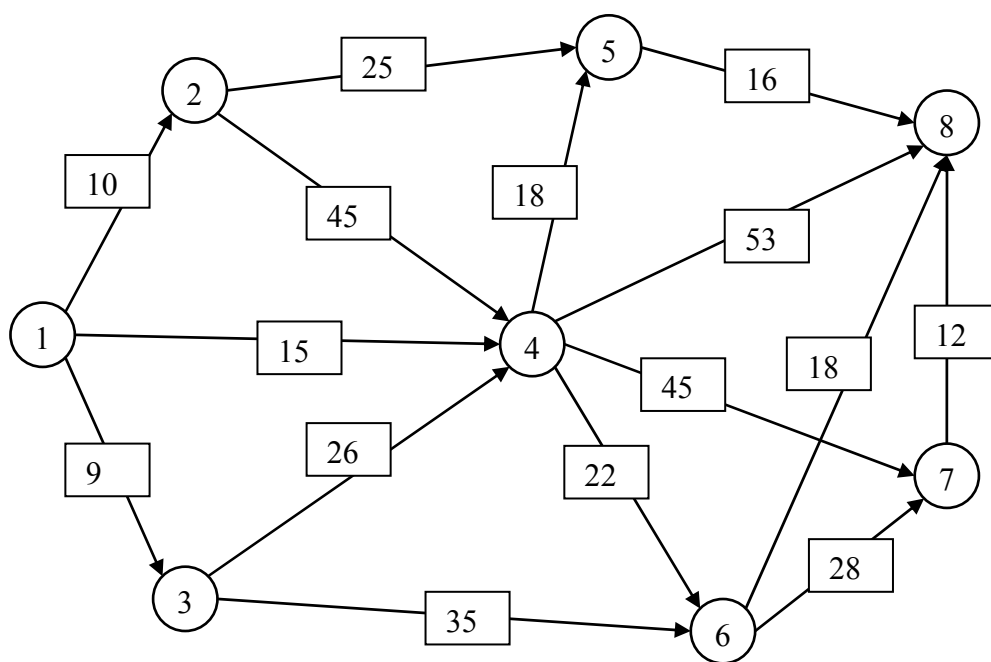
Компания оптовой продажи продуктов питания имеет разветвленную дилерскую сеть. На рис. 6.11 дилерская сеть представлена в виде ориентированного графа с 10 узлами. Вес дуги ориентированного графа – расстояние в тысячах километров. Определить кратчайшие пути между узлами 1 и всеми остальными узлами орграфа.



*Рис. 6.11. Разветвленная дилерская сеть компании оптовой продажи продуктов питания*

### Вариант 12

На рис.6.12 показана транспортная сеть из 8 городов. Расстояние между городами в километрах приведены возле соответствующих дуг графа. Необходимо определить кратчайшие расстояния от города 1 (узел 1) до всех остальных городов (узлов).



*Рис. 6.12. Транспортная сеть из 8 городов*

### Вариант 13

Судоходная компания осуществляет перевозку грузов морскими судами в различные страны мира. Транспортная сеть перевозок представлена на рис. 6.13 в виде ориентированного графа с 9 узлами. Найти кратчайшие расстояния между узлом 1 и всеми остальными узлами графа. Вес дуги – расстояние в милях.

### Вариант 14

Компания AirWays выполняет перевозку пассажиров в 8 городов мира. Транспортная сеть представлена в виде ориентированного графа (рис. 6.14). Вес дуги – расстояние в километрах. Определить кратчайшие пути от узла 1 до всех узлов ориентированного графа.

### Вариант 15

Транспортная компания осуществляет поставку продукции в 10 городов страны. Транспортная сеть представлена на рис. 6.15. Вес дуги – расстояние в километрах. Определить кратчайшие пути между узлом 1 и остальными узлами ориентированного графа.

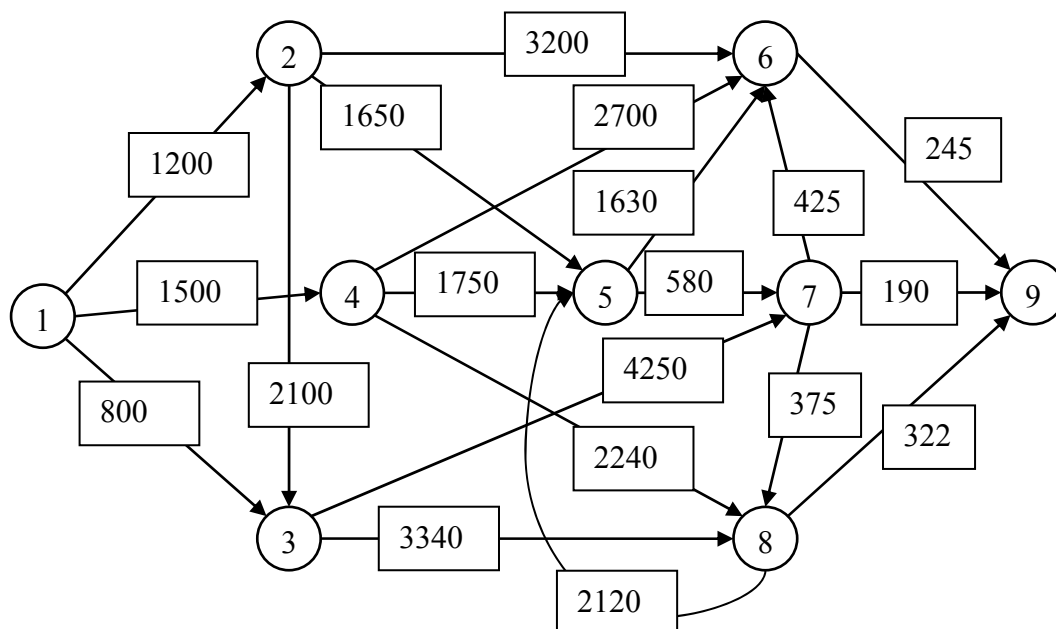


Рис. 6.13. Транспортная сеть перевозок судоходной компании



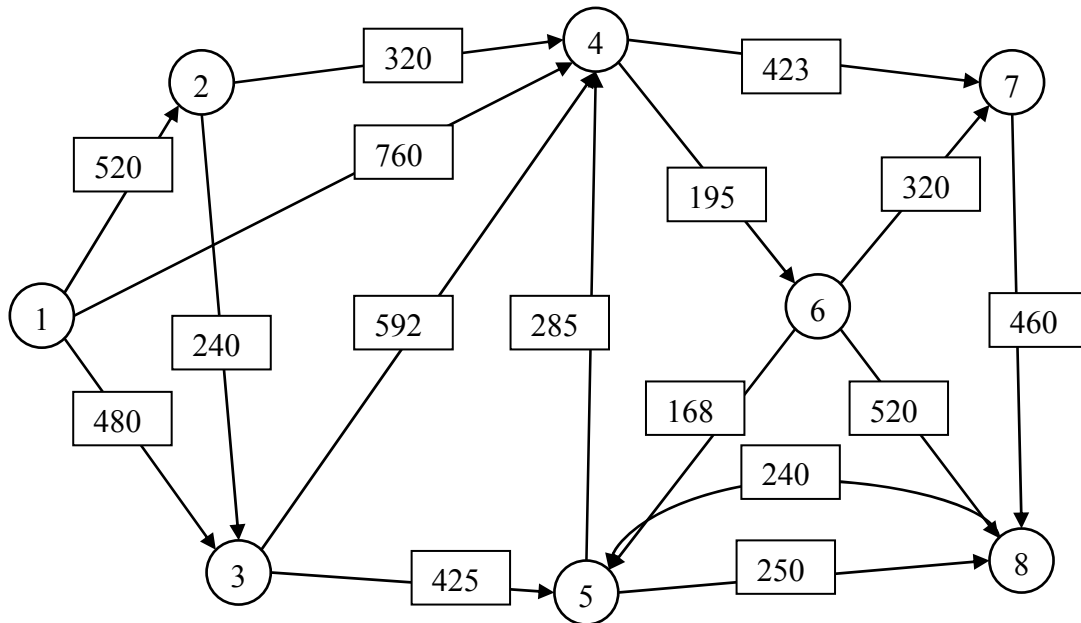


Рис. 6.14. Транспортная сеть компании AirWays

### 6.5.2. Поиск кратчайшего пути. Алгоритм Флойда

#### Вариант 1

Телефонная компания обслуживает 6 удаленных друг от друга районов, которые связаны сетью, показанной на рис. 6.16.

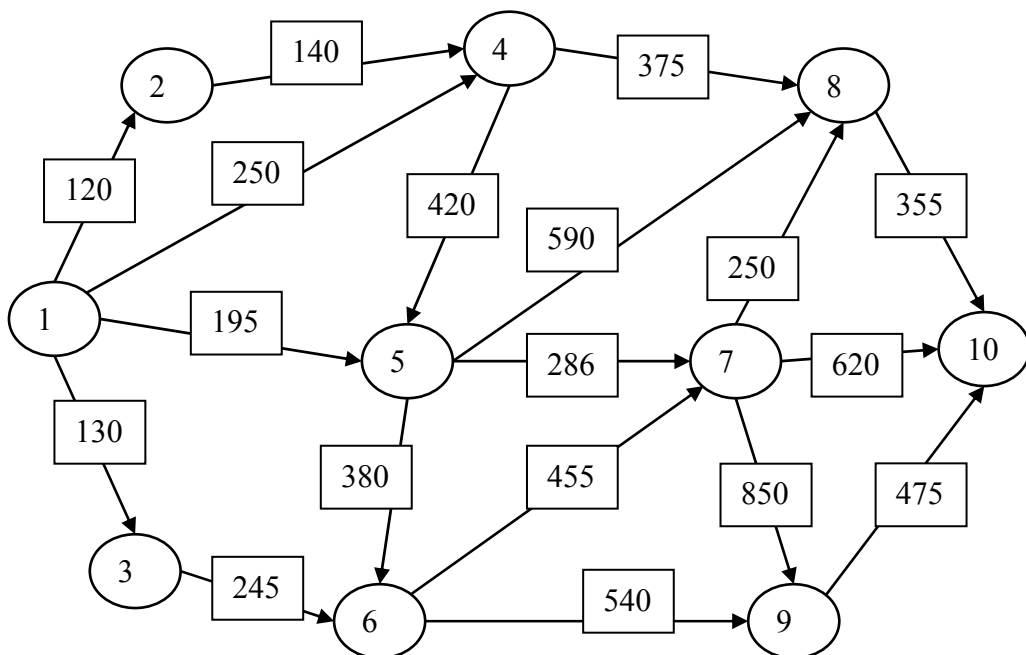
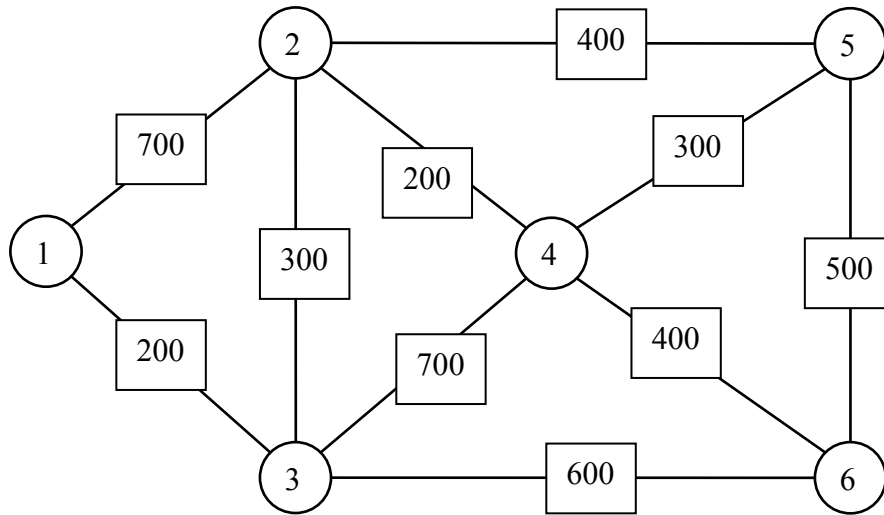
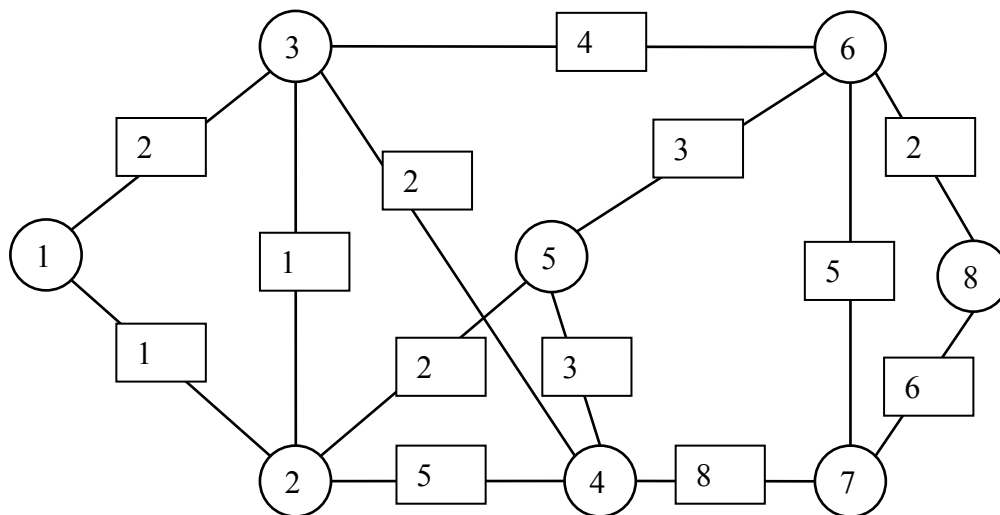


Рис. 6.15. Сеть транспортной компании

Расстояние на схеме сети указано в милях. Компании необходимо определить наиболее эффективные маршруты пересылки сообщений между любыми двумя районами.



*Рис. 6.16. Сеть телефонной компании*



*Рис. 6.17. Транспортная сеть перевозок денежных средств частным охранным предприятием*

## Вариант 2

Частное охранное предприятие осуществляет перевозку денег от ювелирных магазинов, представленных графом с 8 узлами. Расстояние между магазинами указано в километрах. Определить кратчайшие маршруты между ювелирными магазинами (рис. 6.17).

## Вариант 3

Нефтедобывающая фирма BP Sea Company владеет платформами, добывающими нефть в открытом море. Сеть нефтедобывающих платформ представлена на рис. 6.18.

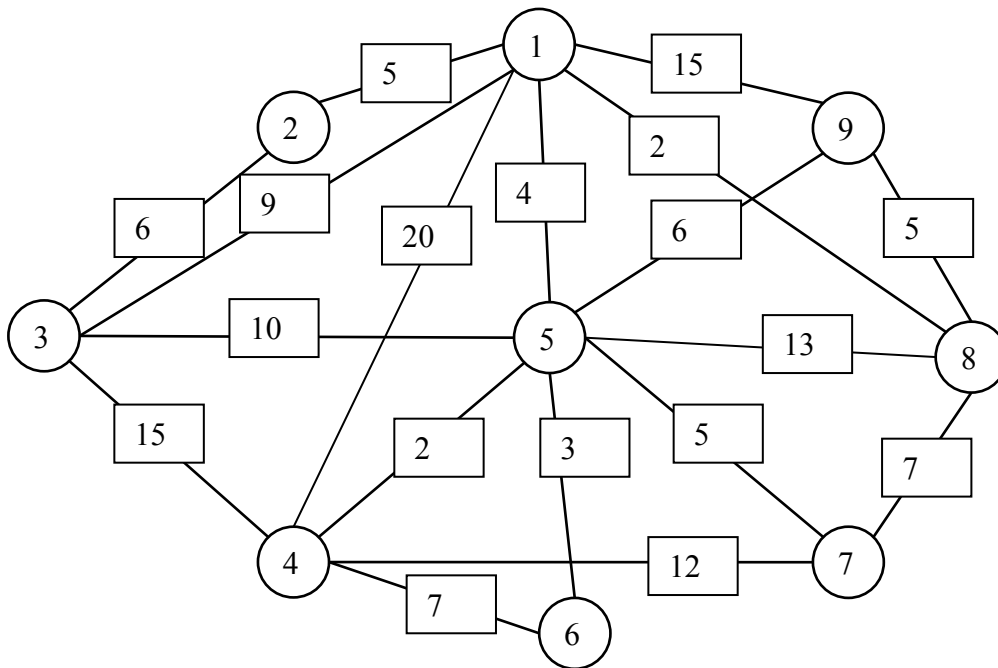


Рис. 6.18. Сеть нефтедобывающих платформ компании BP Sea Company

Расстояние между платформами указано в километрах. Определить кратчайшие пути между платформами.

## Вариант 4

В модульных перевозках грузенные трейлерные платформы перевозятся по железной дороге между железнодорожными терминалами. На рис. 6.19 показаны железнодорожные терминалы США и железнодорожные пути между ними. Расстояние указано в милях. Определить кратчайшие пути между всеми железнодорожными терминалами.

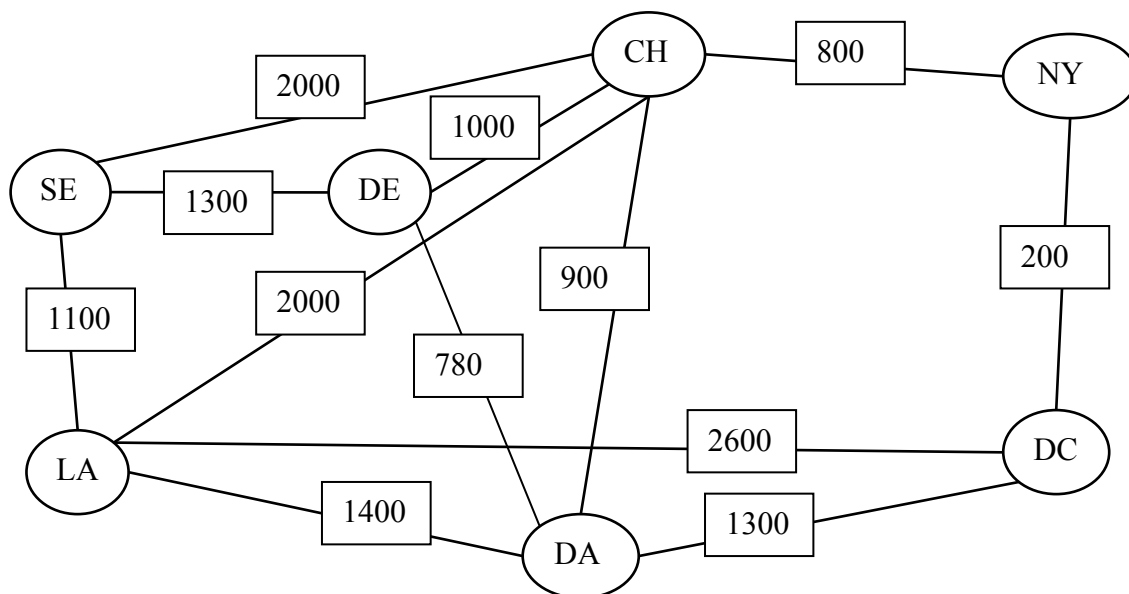


Рис.6.19. Сеть железнодорожных терминалов

### Вариант 5

Торговая фирма производит продажу бытовой техники в 7 городах страны. Сеть филиалов представлена в виде графа на рис. 6.20. Расстояние между городами указано в тысячах километрах. Определить кратчайшие маршруты между городами, в которых расположены филиалы торговой фирмы.

### Вариант 6

Компания грузоперевозок осуществляет доставку товаров в 7 городов страны. Сеть доставки показана на рис. 6.21. Расстояние между узлами графа выражено в тысячах километров. Определить кратчайшие пути между узлами графа.

### Вариант 7

Торговая фирма представлена сетью магазинов, расположенной в городе. Расстояние между магазинами указано в километрах (рис. 6.22). Определить кратчайшие пути между всеми магазинами торговой компании.

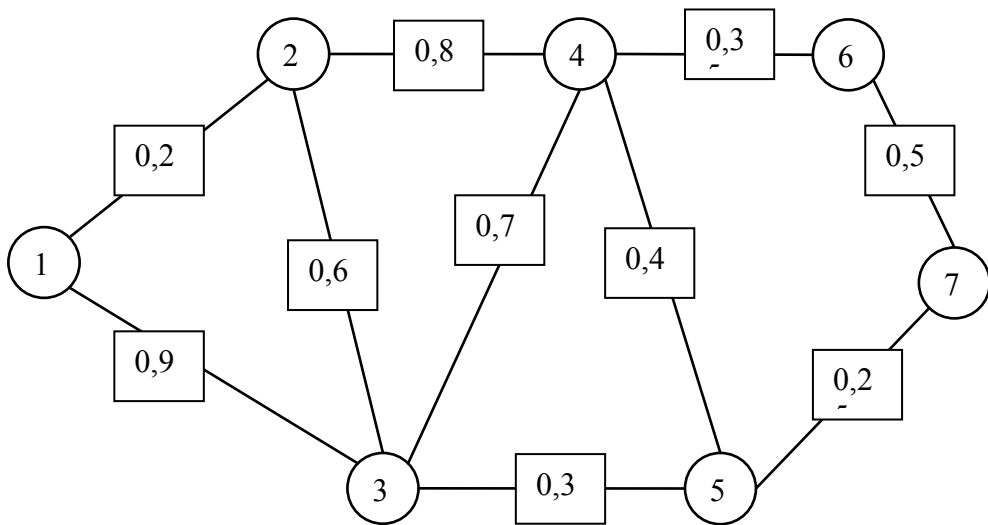


Рис. 6.20. Транспортная сеть филиалов торговой фирмы

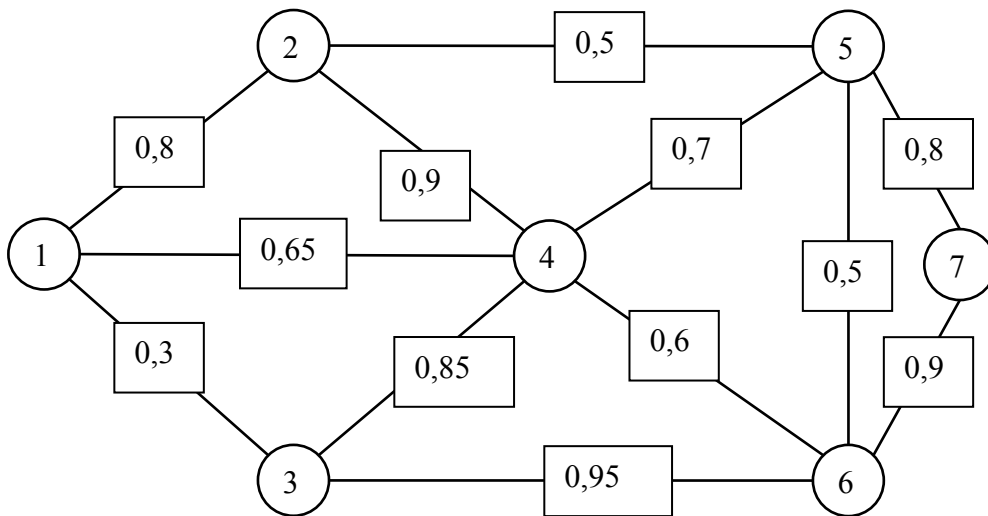


Рис. 6.21. Транспортная сеть доставки грузов компании

### Вариант 8

Авиационная компания осуществляет перевозку пассажиров между 8 городами. Маршруты перевозок представлены в виде графа на рис. 6.23. Расстояние между городами выражено в километрах. Найти кратчайшие пути между всеми городами.

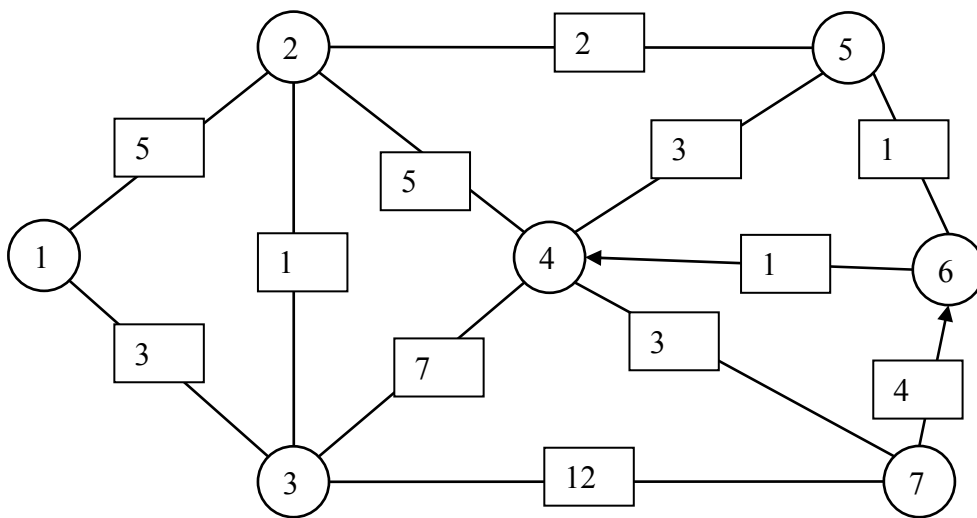


Рис. 6.22. Сеть магазинов торговой фирмы

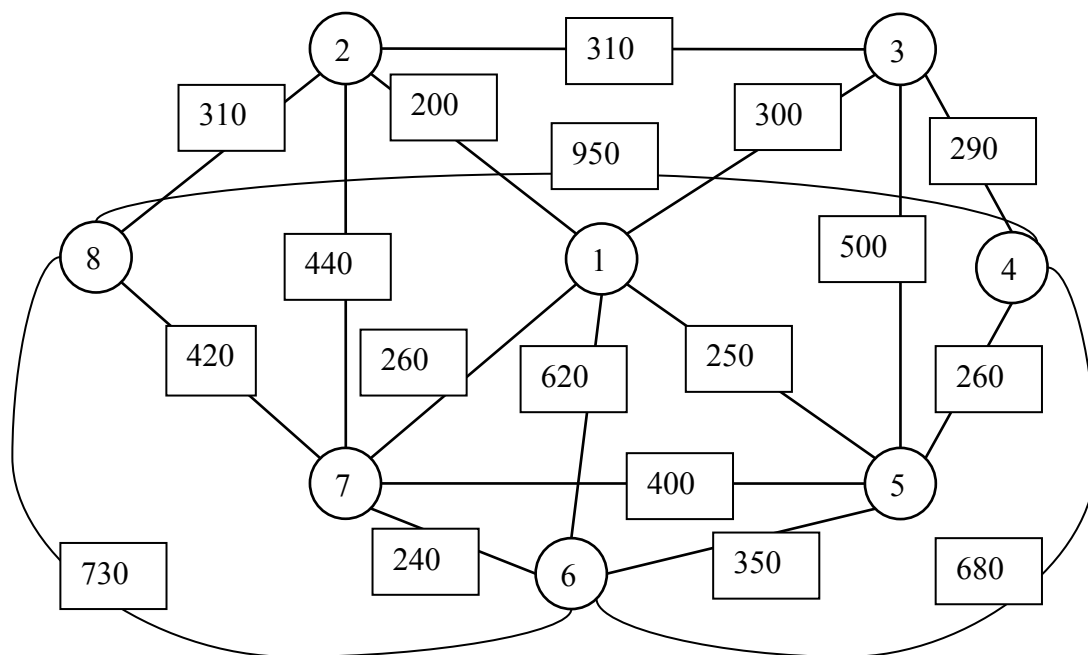


Рис. 6.23. Транспортная сеть авиационной компании

### Вариант 9

Железнодорожная компания осуществляет перевозку пассажиров в 7 городов страны. Сеть перевозок изображена на рис. 6.24. Расстояние между городами указано в километрах. Найти кратчайшие пути между городами.

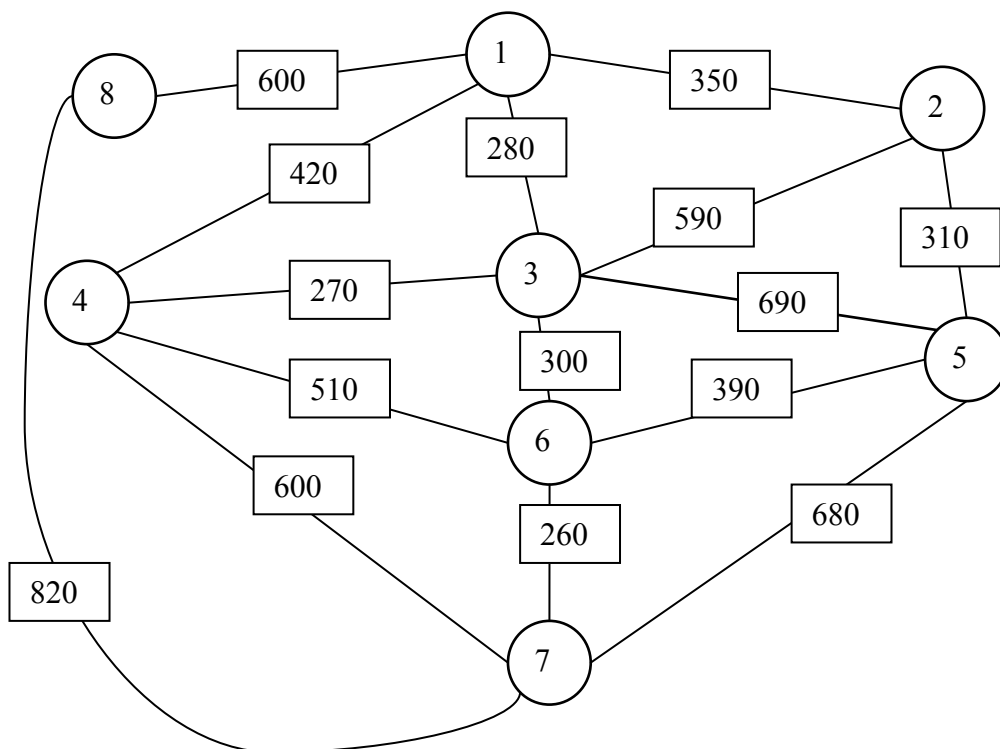


Рис. 6.24. Транспортная сеть железнодорожной компании

### Вариант 10

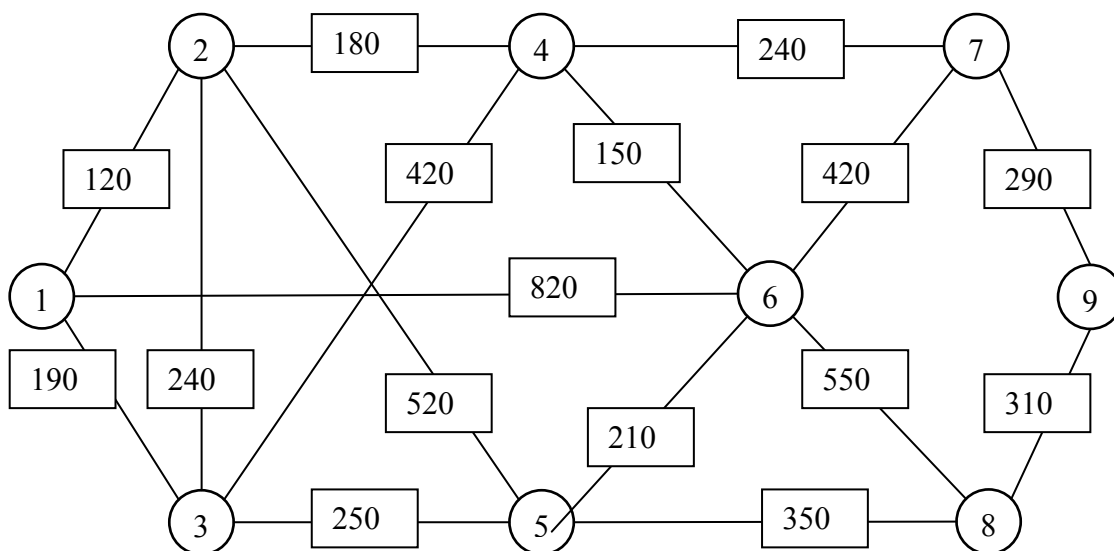


Рис. 6.25. Сеть месторождений добычи газа

Компания осуществляет добычу газа и транспортировку. Сеть месторождений показана на рис. 6.25. Расстояние между месторождениями

указано в километрах. Найти кратчайшие пути между всеми месторождениями по добыче газа компании.

### Вариант 11

Судоходная компания осуществляет перевозку грузов морским транспортом в 8 портов мира. Сеть перевозок грузов судоходной компанией показана на рис. 6.26. Расстояние между портами указано в морских милях. Найти кратчайшие пути между всеми морскими портами, в которые осуществляется доставка груза.

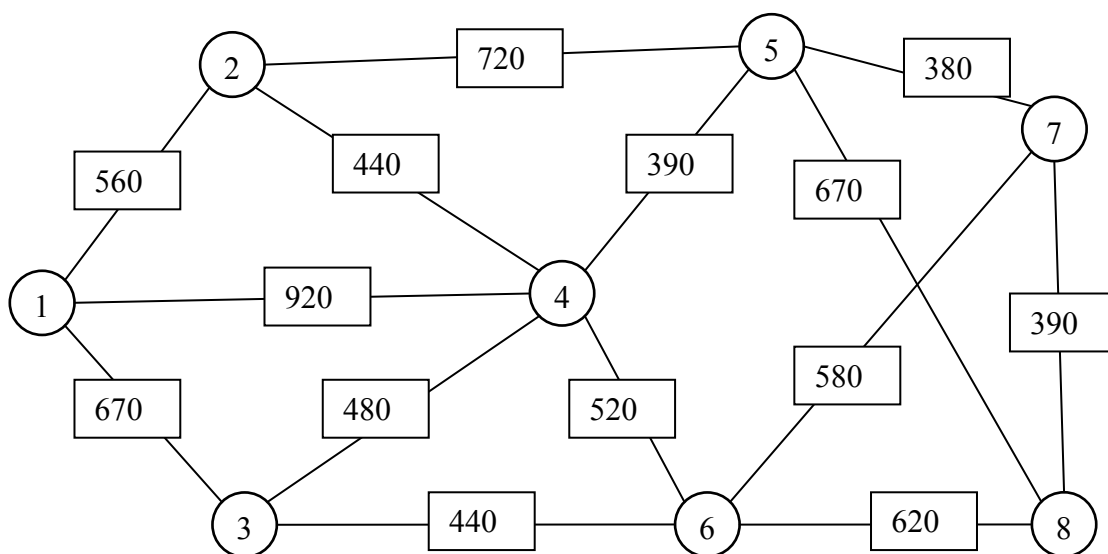


Рис. 6.26. Транспортная сеть перевозок судоходной компании

### Вариант 12

Торговая фирма располагает разветвленной сетью филиалов, представленной в виде графа на рис. 6.27. Расстояние между филиалами выражено в километрах. Найти кратчайшие пути между всеми филиалами.

### Вариант 13

Частное охранное предприятие осуществляет сопровождение ценных грузов между населенными пунктами. Сеть маршрутов изображена в виде графа (рис. 6.28). Расстояние между населенными пунктами (узлами графа) указано в километрах. Найти кратчайшие пути между всеми населенными пунктами.



## Вариант 14

Фирма производит строительство нефтепровода между нефтеперерабатывающими предприятиями. Сеть маршрутов изображена на рис. 6.29 в виде графа.

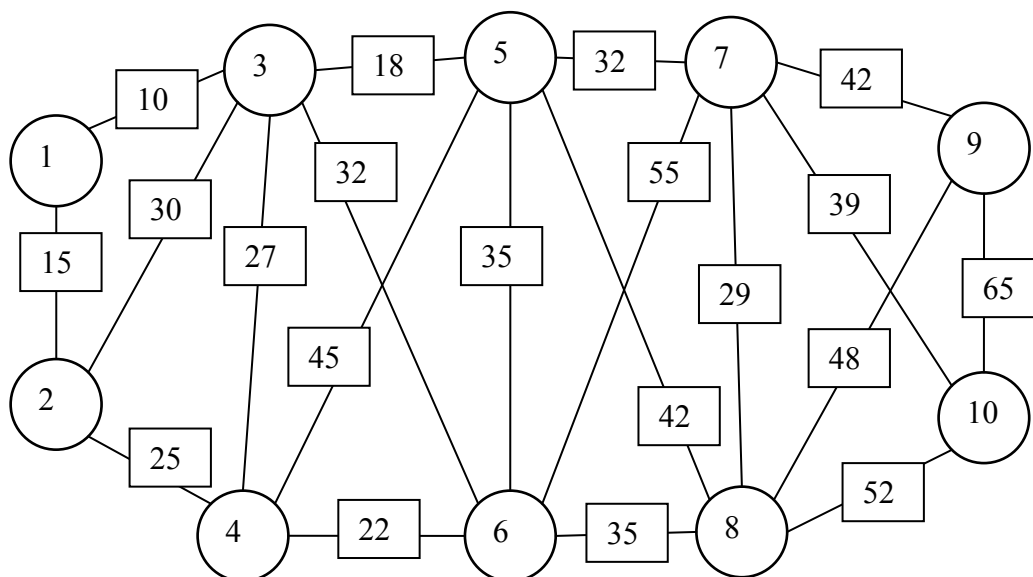


Рис. 6.27. Разветвленная сеть филиалов торговой фирмы

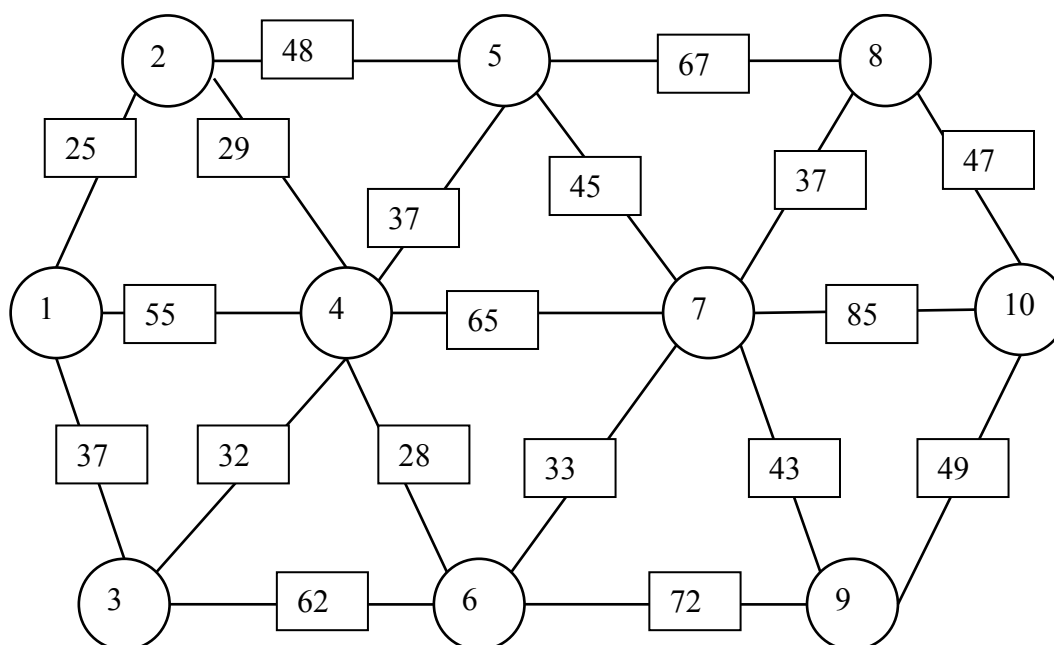


Рис. 6.28. Транспортная сеть перевозки ценных грузов частным охранным предприятием

Расстояние между нефтеперерабатывающими предприятиями (узлами графа) указано в километрах. Найти кратчайшие пути между нефтеперерабатывающими предприятиями

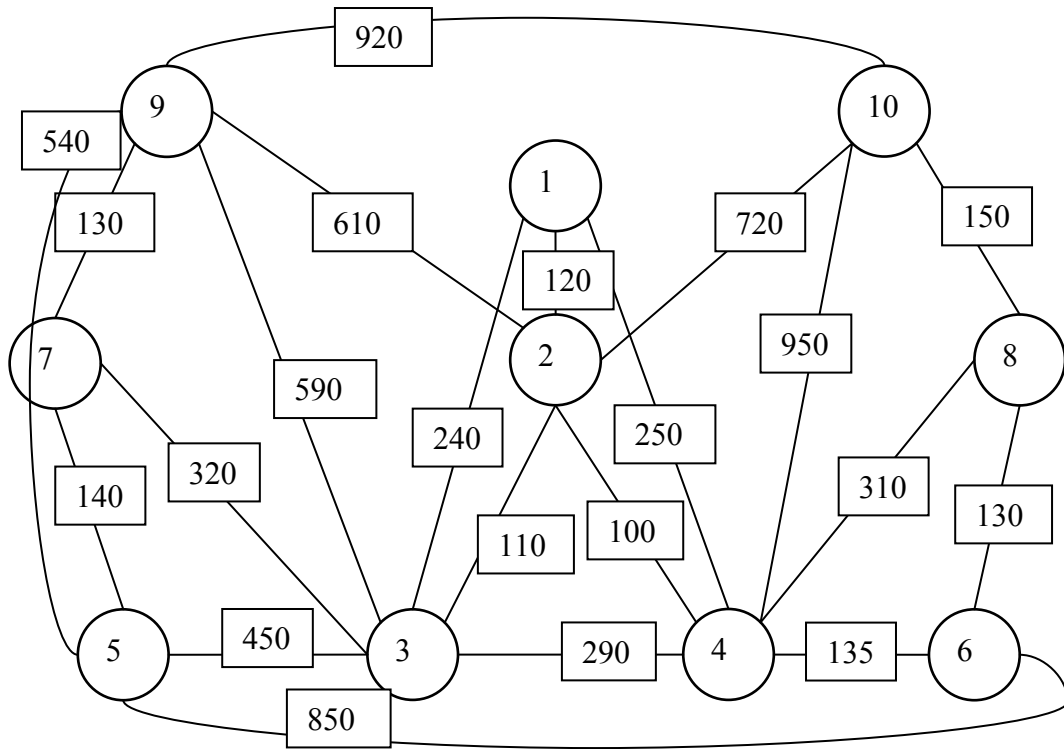


Рис. 6.29. Сеть нефтепроводов между нефтеперерабатывающими предприятиями

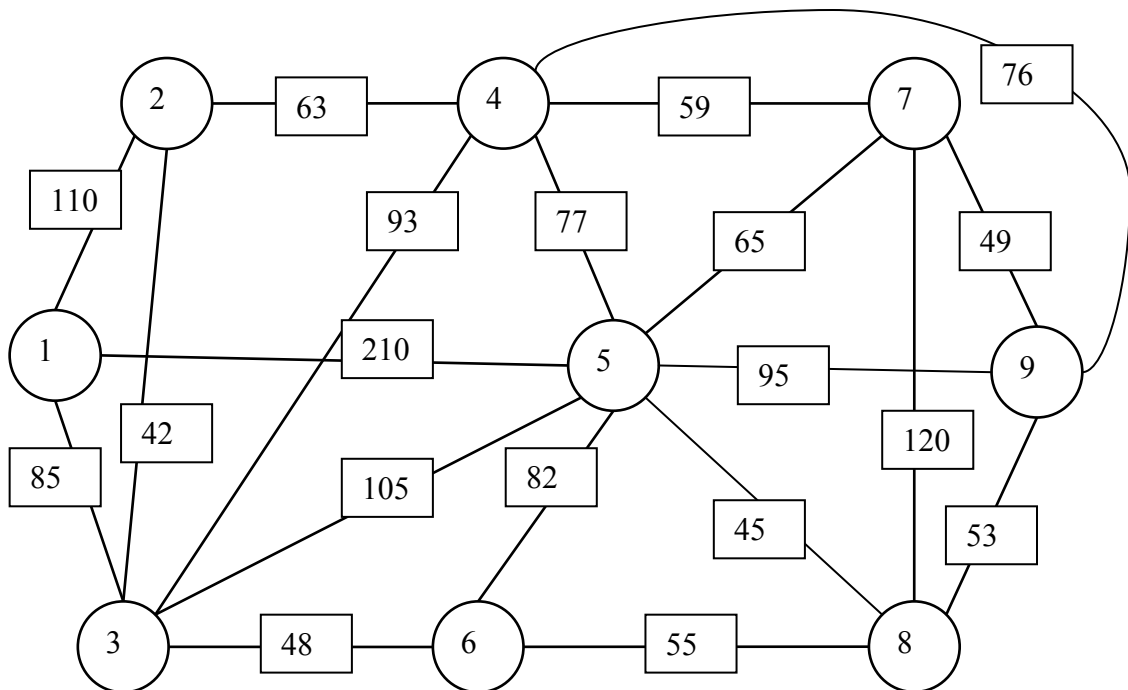


Рис. 6.30. Сеть газопровода между населенными пунктами

## Вариант 15

Строительная фирма производит строительство газопровода между населенными пунктами. Сеть газопровода изображена в виде графа (рис. 6.30). Расстояние между населенными пунктами (узлами графа) указано в километрах. Найти кратчайшие пути между всеми населёнными пунктами.

### 7. Курсовая работа

#### *7.3.1. Назначение, цели и задачи курсовой работы*

Проектируемое приложение курсовой работы предназначено для формализации и анализа качества алгоритмов решения задачи.

**Основная учебная цель** выполнения курсовой работы – закрепление теоретических знаний и приобретение практических навыков в программной реализации и анализе сложных структур данных и алгоритмов их обработки.

**Основные задачи**, решаемые студентом в процессе выполнения курсовой работы:

- программная реализация задачи;
- оценка качества реализованного алгоритма;
- проведение экспериментального исследования алгоритма и анализ его результатов;
- документирование курсовой работы в соответствии требованиями.

#### *7.3.2. Требования к курсовой работе*

##### *7.3.2.1. Требования к функциональным характеристикам*

Проектируемая система должна обеспечивать выполнение следующих основных функций:

- ввод исходных данных задачи;
- расчет параметров;
- оценка стоимости реализации алгоритма по временным и объемным параметрам;
- хранение исходных данных с возможностью их загрузки для повторной обработки;
- хранение результатов решения с возможностью их повторной визуализации;
- вывод результирующих данных;

##### *7.3.2.2. Требования к эксплуатационным характеристикам*

- Модульность.
- Расширяемость.

### *7.3.2.3. Требования к программному обеспечению*

Среда разработки – MS Visual C++ версии не ниже 6.0.

### *7.3.2.4. Требования к содержанию курсовой работы*

К защите курсовой работы должен быть представлен альбом, включающий следующие проектные, программные и эксплуатационные документы:

- Опись альбома.
- Пояснительная записка, включающая разделы:
  - аналитический обзор;
  - описание алгоритмов решения задачи;
  - описание структуры программного комплекса;
  - описание структур данных;
  - описание методики проведения экспериментального исследования;
  - описание и анализ результатов проведенного исследования;
  - выводы по результатам проведенного анализа.
- Спецификация.
- Описание программы.
- Текст программы (на машинном носителе).
- Руководство пользователя.

Пояснительная записка оформляется в соответствии с требованиями методического указания к оформлению документации курсовых и дипломных проектов [14].

### *7.3.3. Варианты заданий курсовой работы*

1. Транспортная задача. Метод потенциалов.
2. Эвристические методы поиска решения на графах. Методы экстремума и минимальной стоимости.
3. Динамическое программирование. Метод обратной прогонки.
4. Сортировка файлов простым слиянием.
5. Сортировка файлов естественным слиянием.
6. Транспортная задача. Распределительный метод.
7. Алгоритм нахождения минимального остовного дерева.
8. Алгоритм определения максимального потока.
9. Алгоритм минимизации стоимости потока в сети с ограниченной пропускной способностью.
10. Алгоритм определения критического пути.
11. Методы поиска решения на графах. Методы поиска в ширину и глубину.
12. Алгоритм поиска оптимального кода. Алгоритм Хаффмана.
13. Динамическое программирование. Метод прямой прогонки.
14. Алгоритм поиска оптимального кода. Алгоритм Шеннона-Фено.

15. Алгоритм сжатия данных. Алгоритм «арифметическое кодирование».
16. Структуры данных и алгоритмы для внешней памяти. Внешние деревья поиска.
17. Алгоритм решения задач выбора. Алгоритм с возвратом.
18. Алгоритм решения задач с выбором. Метод ветвей и границ.
19. Остовное дерево наименьшей стоимости (минимального веса). Алгоритм Прима.
20. Остовное дерево наименьшей стоимости (минимального веса). Алгоритм Крускала.
21. Алгоритм разбиения. Задача определения максимальной суммы непрерывной подпоследовательности.
22. Алгоритмы внешней сортировки. Многофазная сортировка.
23. Алгоритмы внешней сортировки. Каскадная сортировка.
24. Алгоритмы на графах. Задача поиска кратчайшего отрицательно взвешенного пути при одном источнике.
25. Алгоритмы на графах. Задача поиска кратчайшего пути в ациклическом графе при одном источнике.

## ЗАКЛЮЧЕНИЕ

В методических указаниях приведен теоретический материал по динамическим структурам данных и представлены варианты заданий для выполнения практических и контрольных работ.

Особенностями динамических структур данных являются:

- блоки в динамической памяти, связаны друг с другом с помощью указателей;
- динамические структуры различаются способами связи отдельных элементов и допустимыми операциями над ними;
- элемент динамической структуры данных состоит из информационных полей и полей указателей;
- распространенными структурами являются линейный список, стек, очередь и бинарное дерево;
- для стека определены операции помещения элемента в вершину и выборки элемента из вершины;
- для очереди определены операции помещения элемента в конец очереди и выборка элемента из ее начала;
- допускается вставлять и удалять элементы в произвольное место линейного списка;
- бинарное дерево состоит из узлов, каждый из которых содержит данные и не более двух ссылок на различные бинарные деревья. На каждый узел имеется ровно одна ссылка;
- каждый узел характеризуется уникальным ключом;
- допускается вставлять и удалять элементы в произвольное место дерева;
- если для каждого узла все ключи его левого поддерева меньше ключа этого узла, а все ключи его правого поддерева – больше, то такое дерево называется деревом поиска. В дереве поиска можно найти элемент по ключу, двигаясь от корня и переходя на левое или правое поддерево в зависимости от значения ключа в каждом узле;
- динамические структуры в некоторых случаях более эффективно реализовывать с помощью массивов.

Выполнение практических и контрольных работ позволяют закрепить знания по дисциплине «алгоритмы и структуры данных».

Изучение структур данных важно для освоения библиотеки стандартных шаблонов (STL). STL составляет основную часть Стандартной библиотеки. STL упаковывает структуры данных в классы, оформленные в виде шаблонов.

## СПИСОК ЛИТЕРАТУРЫ

1. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы: Учебное пособие/Пер. с англ. - М.: Издательский дом «Вильямс», 2000. – 432 с.
2. Каррано Фрэнк М. Абстракция данных и решение задач на C++. Стены и зеркала. – М.; СПб; Киев: «Вильямс», 2003. – 848 с.
3. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Анализ и построение. – М.: «БИНОМ», 2000. – 960 с.
4. Сэдживик Р. Фундаментальные алгоритмы на C++. – М.: «DiaSoft», 2001. – 688 с. – Ч. 1-5.
5. Кубенский А.А. Структуры и алгоритмы обработки данных. Объектно-ориентированный подход и реализация на C++. – СПб.: БХВ-Петербург, 2004. – 466 с.
6. Хусаинов Б.С. Структуры и алгоритмы обработки данных. Примеры на языке Си: Учебное пособие. – Финансы и статистика, 2004. – 464 с.
7. Труб И.И. Объектно-ориентированное моделирование на C++: Учебный курс. – СПб.: Питер, 2006. – 411 с.
8. Павловская Т.А. C/C++. Программирование на языке высокого уровня. – СПб.: Питер, 2006. – 461 с.
9. Павловская Т.А., Щупак Ю.А. C/C++. Структурное программирование: Практикум. – СПб.: Питер, 2007. – 239 с.
10. Давыдов В.Г. Программирование и основы алгоритмизации: Учебное пособие. – М.: Высш. шк., 2003. – 447 с.
11. Давыдов В.Г. Технологии программирования. C++. – СПб.: БХВ – Петербург, 2005. – 672 с.
12. Дейтл Х.М., Дейтл П. Дж. Как программировать на C++/ Пер. с англ. – 5-е изд-е. - М.: ООО «Бином-Пресс», 2008. - 1456 с.
13. Таха Хемди А. Введение в исследование операций / Пер. с англ.- 7-е изд-е. – М.: Издательский дом «Вильямс», 2005. – 912 с.
14. Дик Д.И. Требования к оформлению текстовой документации курсовых и дипломных проектов (работ): Методические указания для студентов направлений (специальностей) 230000 (230105), 090000 (090105). – Курган: Из-во КГУ, 2008. - 40 с.

Семахин Андрей Михайлович

## АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

Методические указания  
к выполнению лабораторных и курсовых работ  
для студентов направления (специальности)  
231000.62 – «Программная инженерия»

Редактор А.С. Мокина

---

Подписано в печать	Формат 60x84 1/16	Бумага тип №1
Печать трафаретная	Усл. печ. л. 4,0	Уч.–изд. л. 4,0
Заказ	Тираж 30	Цена свободная

---

РИЦ Курганского государственного университета.  
640669, г. Курган, ул. Гоголя, 25.  
Курганский государственный университет.