АЛГОРИТМЫ ОБХОДА ГРАФОВ

Изучение основных понятия теории графов и некоторых алгоритмов обработки графов иприобретение практических навыков по созданию консольных приложений для решения «транспортных» задач с использованием графов.

6.2 Теоретические сведения

6.2.1 Стягивающие деревья. Основные понятия

В теории графов существует два простых определения понятия дерева. Не разбирая теоремы, доказывающие справедливость этих определений приведем их текст. Первое определение утверждает, что деревом является связанный граф, в котором число дуг на единицу меньше числа вершин.

Второе определение более очевидное – деревом называется связанный граф без циклов.

Дерево, полученное из графа путем удаления некоторых ребер, называется стягивающим деревом этого графа или каркасом.

Например, на рисунке 6.1 представлен граф типа «звезда», который может быть представлен несколькими стягивающими деревьями. Один из вариантов стягивающего дерева также изображен на рисунке 6.1.

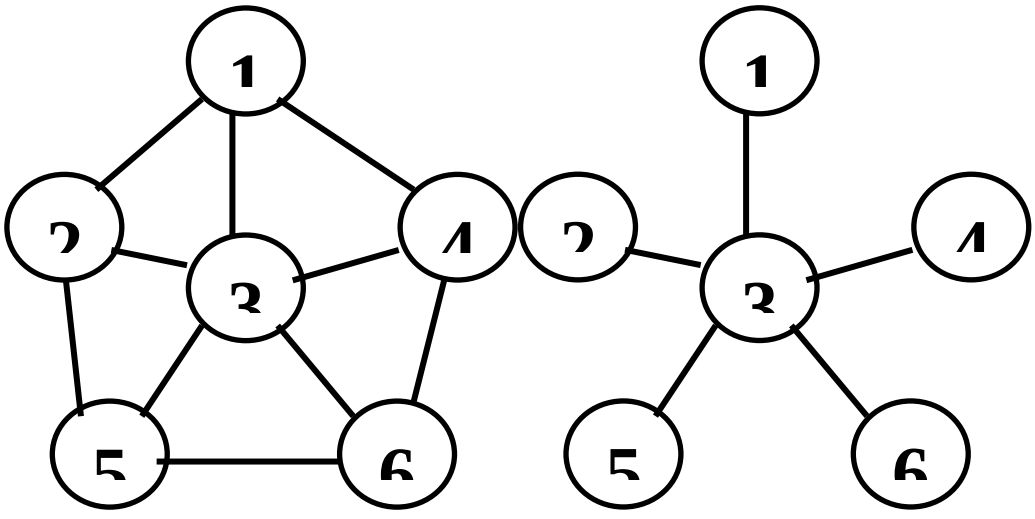

Рисунок 6.1 – Связанный граф и его стягивающее дерево.

В теории графов неиспользованные ребра (удаленные ребра графа из которого получено стягивающее дерево) называются хордами.

Если к стягивающему дереву добавить произвольную хорду, то полученный граф содержит в точности один цикл.

На этой идее строятся алгоритмы получения всего множества циклов графа, что часто имеет прикладной характер, например, при анализе электрических цепей по закону Кирхгофа.

6.2.2 Алгоритм построения стягивающего дерева

Существует множество различных алгоритмов нахождения стягивающих деревьев.

Рассмотрим алгоритм нахождения стягивающего дерева, в котором использован известный Вам алгоритм Дейкстры – нахождения минимальных маршрутов от заданной вершины до остальных вершин графа.

Это возможно, так как существует теорема, доказывающая, что «набор» минимальных маршрутов от заданной вершины до остальных вершин графа есть стягивающее дерево.

Напомню идею алгоритма нахождения минимального маршрута между вершинами графа. Например, существуют три смежные вершины 1,2 и 3. Расстояние между вершинами 1 - 2 равно 5, вершинами 2 – 3 равно 3, а вершинами 3 – 1 равно 1. При выборе маршрута между вершинами 1 – 2 графа используется маршрут 1 – 3 – 2. Ребро 1 – 2 удаляется из графа.

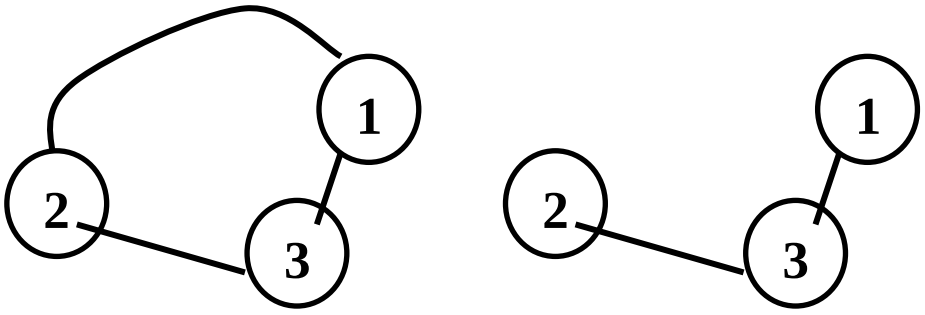

Рисунок 6.2 – Идея алгоритма нахождения минимального маршрута

Использование приведенного алгоритма позволяет строить стягивающие деревья. Так как «перебор» вершин графа в алгоритме осуществляется в порядке возрастания вершин (использован цикл for), то при «равенстве» двух маршрутов в стягивающем дереве остается первый вариант.

В качестве исходного графа для программной реализации задачи построения стягивающего дерева связанного графа, выбран граф, для которого уже существует программная реализация в визуальной среде программирования Delphi (граф изображен на рисунке 6.3).

Алгоритм построения стягивающего дерева использует алгоритм нахождения минимальных маршрутов Дейкстры. Отличие от ранее рассмотренной программой реализации этого алгоритма заключается в том, что можно задавать значение начальной вершины.

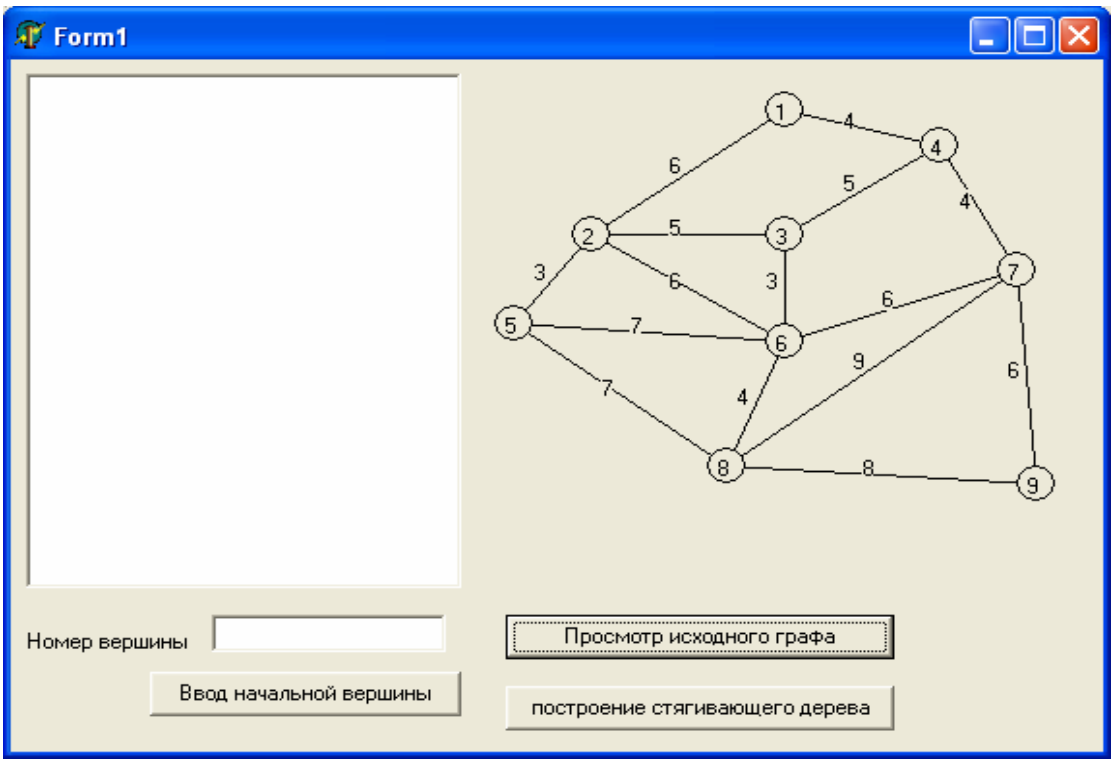

Рисунок 6.3 – Исходный связанный граф

6.2.3 Алгоритм Дейкстры

Алгоритм Дейкстры требует использования трех массивов, размерность которых соответствует количеству вершин графа.

Первый массив, массив «постоянных» вершин, будет хранить минимальные маршруты от заданной вершины графа до всех остальных его вершин. Первоначально в этот массив записывается номер начальной вершины (вершины, от которой будем находить минимальные расстояния для всех других вершин графа). Название массива соответствует названию выбранных вершин в алгоритме Дейкстры, согласно которому сначала все вершины графа объявляются «временными», а выбранные вершины называются «постоянными». Начальная вершина объявляется «постоянной».

Второй массив предназначен для хранения минимальных расстояний от заданной вершины графа до всех остальных его вершин. Первоначально в него переписывается строка матрицы смежности, соответствующая номеру выбранной вершины.

В третьем массиве логического типа отмечаются выбранные (постоянные) вершины графа. Первоначально «постоянной» вершиной графа является только начальная вершина.

Далее выбирается «временная» вершина, до которой расстояние от начальной (постоянной) вершины минимально. Эта вершина объявляется вершиной k. Проверяются все маршруты от «постоянной» вершины до всех «временных» вершин графа как непосредственно, так и через вершину k.Минимальные расстояния заносятся вво второй массив, а в первый массив,

в соответствующую позицию, записывается k, если маршрут минимального расстояния проходил через вершину k.

Далее вершина k объявляется «постоянной» (это фиксируется в третьем массиве) и алгоритм поиска следующей вершины графа повторяется относительно новой «постоянной» вершины.

6.3 Пример выполнения задания на лабораторную работу

Задача 6.1 Для связанного графа, изображенного на рисунке 6.3 построить стягивающее дерево. Номер начальной вершины графа задавать в режиме диалога. Стягивающее дерево представить списком маршрутов от заданной вершины до остальных вершин графа.

Исходный код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
classProgram
 {
publicstaticint[,] a = newint[9, 9]
   {{   0,   6,1000,   4,1000,1000,1000,1000,1000},
    {   6,   0,   5,1000,   3,   6,1000,1000,1000},
    {1000,   5,   0,   5,1000,   3,1000,1000,1000},
    {   4,1000,   5,   0,1000,1000,   4,1000,1000},
    {1000,   3,1000,1000,   0,   7,1000,   7,1000},
    {1000,   6,   3,1000,   7,   0,   6,   4,1000},
    {1000,1000,1000,   4,1000,   6,   0,   9,   6},
    {1000,1000,1000,1000,   7,   4,   9,   0,   8},
    {1000,1000,1000,1000,1000,1000,   6,   8,   0}};

publicstaticint[] d = newint[10];
publicstaticint[] post = newint[10];
publicstaticbool[] t = newbool[10];
publicstaticint i, j, p, k, minras, begver;

publicstaticvoid poisk()
{
string buf;
//Ввод значения переменной a в режиме диалога
Console.WriteLine("Введите номер вершины графа целое значение 0
- 8");
buf = Console.ReadLine();
   begver = Convert.ToInt32(buf);
//начальные установк
for (i = 0; i < 9; i++)
{
     post[i] = begver;
     t[i] = true;
```

```
d[i] = a[begver,i];
    }
   t[begver] = false;
   post[begver] = 0;
   minras = 0;
for (i = 0; i < 8; i++)
    {
// поисквершины k
     minras = 1000;
for (j = 0; j < 9; j++)
if ((t[j] == true) && (minras > d[j]))
{
        minras = d[j]; k = j;
}
// поиск маршрутов и минимальных расстояний через вершину k
t[k] = false;
for (j = 0; j < 9; j++)
if ((t[j] == true) && (d[j] > d[k] + a[k,j]))
      {
       d[j] = d[k] + a[k,j];
       post[j] = k;
      }
    }
  }


publicstaticvoid print()
{
Console.WriteLine("Печатаемматрицусмежности : ");
for (i = 0; i < 9; i++)
    {
for (j = 0; j < 9; j++)
Console.Write("\t" + a[i, j]);
Console.WriteLine();
    }
Console.WriteLine();
// печатьномероввершинграфа
for (i = 0; i < 9; i++) Console.Write("\t" + i);
Console.WriteLine();
// печать массива минимальных расстояний
for (i = 0; i < 9; i++) Console.Write("\t" + d[i]);
Console.WriteLine();
// печатьмассивамаршрутов
for (i = 0; i < 9; i++) Console.Write("\t" + post[i]);
Console.WriteLine();
Console.WriteLine();
int dlin;
// печать маршрутов стягивающего дерева
for (i = 0; i < 9; i++)
    {
     dlin = d[i];
Console.Write("Путь  №  {0}-{1}  длина  пути  =  {2}\t\t{3}",  i,
begver, dlin,i);
```

```
p = i;
if (i != begver)
      {
do
        {
         p = post[p];
if (p!=0) Console.Write("\t" + p);
}
while (p != 0);
Console.WriteLine();
      }
elseConsole.WriteLine();
    }
  }

staticvoid Main()
  {
   poisk();
   print();
Console.ReadLine();
  }
 }
}
```

Работапрограммы:

Введите номер вершины графа целое значение 0 - 8

4

Печатаем матрицу смежности :

0	6	1000	4	1000	1000	1000	1000	1000
6	0	5	1000	3	6	1000	1000	1000
1000	5	0	5	1000	3	1000	1000	1000
4	1000	5	0	1000	1000	4	1000	1000
1000	3	1000	1000	0	7	1000	7	1000
1000	6	3	1000	7	0	6	4	1000
1000	1000	1000	4	1000	6	0	9	6
1000	1000	1000	1000	7	4	9	0	8
1000	1000	1000	1000	1000	1000	6	8	0

0	1	2	3	4	5	6	7	8
9	3	8	13	0	7	13	7	15
1	4	1	2	0	4	5	4	7

Путь № 0-4 длина пути = 9	0	1	4	
Путь № 1-4 длина пути = 3	1	4		
Путь № 2-4 длина пути = 8	2	1	4	
Путь № 3-4 длина пути = 13	3	2	1	4
Путь № 4-4 длина пути = 0	4			
Путь № 5-4 длина пути = 7	5	4		
Путь № 6-4 длина пути = 13	6	5	4	

Путь № 7-4 длина пути = 7 7 4
Путь № 8-4 длина пути = 15 8 7 4

6.4 Домашнее задание на лабораторную работу

Разработать программу для реализации алгоритма обхода графа в «ширину».Использоватьсвойграф.Количество вершин не менее 10.

6.5 Индивидуальные задания для СРС

6.5.1 Создать граф – авиационных перевозок, узлы которого дополнительно включают символьное поле – название областных центров республики. Предусмотреть поиск минимального маршрута перемещения от заданного областного центра до всех остальных центров республики.
6.5.2 Создать свой граф не менее 10 вершин, узлы которого соответствуют некоторой электронной схеме без активных элементов. Найти сопротивление каждой цепей между узлами схемы, заданными в режиме диалога.
6.5.3 Схему автобусных маршрутов города представить структурой типа граф. Узлы структуры соответствуют остановкам автобусных маршрутов и дополнительно включают название остановок. Предусмотреть просмотр номеров маршрутов по названию остановки.
6.5.4 Создать граф имен студентов группы (допускается использование одинаковых имен). Предусмотреть поиск студентов по имени заданному в режиме диалога.
6.5.5 Генеалогическое дерево некоторого рода представлено графом не более 12 вершин. Узел каждой вершины графа дополнительно включает пол представителя рода. Организовать поиск и печать всех особ женского пола с помощью обхода графа в «глубину».
6.5.6 Четыре трамвайных маршрута города представлены структурой типа граф. Узлы структуры соответствуют остановкам трамвайных маршрутов и дополнительно включают название остановок. Для двух названий остановок, введенных в режиме диалога, найти минимальный маршрут перемещения от первой остановки до второй (по минимальной сумме расстояний пройденных остановок).
6.5.7 Генеалогическое дерево некоторого рода представлено графом не более 14 вершин. Узел каждой вершины графа дополнительно включает имя представителя рода. Организовать поиск наиболее часто встречающегося мужского и женского имени.
6.5.8 Четыре трамвайных маршрута города представлены структурой типа граф. Узлы структуры соответствуют остановкам трамвайных маршрутов и дополнительно включают название остановок. Для двух названий остановок, введенных в режиме диалога, найти минимальный маршрут перемещения от первой остановки до второй (по количеству пройденных остановок).
6.5.9 Генеалогическое дерево некоторого рода представлено графом не более 15 вершин. Узел каждой вершины графа дополнительно включает основной

вид деятельности представителя рода и время его работы по этой профессии. Организовать поиск профессии с максимальным временем работы.
6.5.10 Четыре трамвайных маршрута города представлены структурой типа граф. Узлы структуры соответствуют остановкам трамвайных маршрутов и дополнительно включают название остановок. Напечатать названия остановок города в порядке убывания числа маршрутов, проходящих через эти остановки.
6.5.11 Иерархическая структура каталогов диска С компьютера представлена структурой типа граф, узлы которого соответствуют папкам каталога и дополнительно включают название папок. Определить, есть ли на диске одинаковые папки. Напечатать их и отобразить путь к ним от корневого каталога.
6.5.12 Создать свой граф не менее 10 вершин, узлы которого дополнительно включают символьное поле. Разработать алгоритм обхода графа в «глубину» только по «гласным» вершинам графа(дополнительное символьное поле содержит «гласный» символ).
6.5.13. Иерархическая структура каталогов диска С компьютера представлена структурой типа граф, узлы которого соответствуют папкам каталога и дополнительно включают название папок. Определить, сколько раз на диске встречается папка с названием Games.
6.5.14 Создать свой граф не менее 10 вершин, узлы которого дополнительно включают символьное поле. Разработать алгоритм обхода графа в «глубину» только по «согласным» вершинам графа (дополнительное символьное поле содержит «согласный» символ).
6.5.15 Схему автобусных маршрутов района области представить структурой типа граф (не менее 10 вершин). Узлы структуры соответствуют названиям поселков района. Предусмотреть просмотр номеров маршрутов по названию остановки. Для названия поселка, введенного в режиме диалога, найти минимальный маршрут перемещения от районного центра до этого поселка (по минимальной сумме расстояний пройденных остановок).
6.5.16 Водопроводная сеть микрорайона города представлена ориентированным графом не менее 15 вершин, а дуги – пропускной способности участка сети (количество воды, подаваемое в секунду). Определить какое максимальное количество воды можно подавать в узлы A, B и C из некоторого узла X. Значение всех узлов задаются в режиме диалога. Использовать равномерное распределение подаваемой воды между принимающими узлами (если это могут обеспечить подводящие дуги).
6.5.17 Создать свой граф не менее 10 вершин, узлы которого соответствуют некоторой электронной схеме без активных элементов. Найти сопротивление цепей между узлами схемы, заданными в режиме диалога (учитывать параллельное и последовательное соединение схемы).
6.5.18 Лабиринт представлен графом не менее 16 вершин, где вершины соответствуют перекресткам или тупикам. Известны узлы входа и выхода лабиринта. Найти минимальный маршрут прохождения лабиринта.

6.5.19 Четыре трамвайных маршрута города представлены структурой типа граф. Узлы структуры соответствуют остановкам трамвайных маршрутов и дополнительно включают название остановок. Для двух названий остановок, введенных в режиме диалога, найти минимальный маршрут перемещения от первой остановки до второй (по количеству совершаемых пересадок, но не количеству остановок). Напечатать путь перемещения с указанием названий остановок и номеров маршрутов трамваев, на которые необходимо садится или пересаживаться.
6.5.20 Четыре трамвайных маршрута города представлены структурой типа граф. Узлы структуры соответствуют остановкам трамвайных маршрутов и дополнительно включают название остановок. Для двух названий остановок, введенных в режиме диалога, найти минимальный маршрут перемещения от первой остановки до второй (по количеству остановок). Напечатать путь перемещения с указанием названий остановок и номеров маршрутов трамваев, на которые необходимо садится или пересаживаться.

6.6 Контрольные вопросы для защиты отчета на СРСП

6.6.1 Понятие графа. Примеры.
6.6.2 Понятие инцидентности. Пример.
6.6.3 Понятие смежности. Пример.
6.6.4 Понятие пути графа. Какой путь графа называется простым? Пример.
6.6.5 Какой граф называется связанным? Пример.
6.6.6 Как в памяти ЭВМ можно разместить информацию о графе?
6.6.7 Объясните представление графа в памяти ЭВМ в виде списков смежных вершин.
6.6.8 Алгоритм поиска минимальных расстояний между вершинами графа.
6.6.9 Алгоритм нахождения минимальных маршрутов между вершинами графа. 6.6.10 Алгоритм обхода графа в «глубину».
6.6.11 Алгоритм обхода графа в «ширину».
6.6.12 Алгоритм построения «стягивающих» деревьев графа.
6.6.13 Алгоритм нахождения всех маршрутов между двумя заданными вершинами графа.
6.6.14 Зачем в алгоритме поиска всех циклов графа просмотренным вершинам графа «возвращается» свойство новой вершины?
6.6.15 Объясните структуру стека, которая используется в алгоритме поиска всех маршрутов графа между двумя заданными вершинам.