МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеський національний політехничний університет

Інститут Комп'ьютерних Систем

Кафедра Інформаційних Систем

Отчет по лабораторной работе №5

«Графы: способы их хранения и обхода (в ширину и в глубину). Проверка графа на двудольность, поиск циклов и топологическая сортировка графа»

Выполнил:

студент группы АД-171

Поликарпов А.В.

Проверил:

Шибаева Н. О.

Одесса 2019

**Цель :** Приобретение практических навыков программирования с использованием графов. Способы представления графов и работа с графами.

**Вариант 16** Имеется N городов. Для каждой пары городов (I,J) можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги A(I,J). Вне городов дороги не пересекаются.

Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей B[1:N,1:N], где B[I,J]=1 тогда и только тогда, когда дорогу, соединяющую города I и J, следует строить.

Код

[#include](tg://search_hashtag?hashtag=include) <stdio.h>  
[#include](tg://search_hashtag?hashtag=include) <conio.h>  
[#define](tg://search_hashtag?hashtag=define) max\_v 100000  
[#define](tg://search_hashtag?hashtag=define) min(x,y) ((x)<(y)?(x):(y))  
  
int mass[100][100]; // матрица смежностей  
int i,j,N,M,W;  
int a,b,c;  
int start,end;  
int D[100]; // массив, который хранит минимальный путь  
int used[100];  
  
int short\_way (int start, int end);  
  
int main ()  
{  
  
  
FILE \*inp = fopen("Gf.txt","rt");  
  
fscanf (inp,"%d %d", &N, &M); // количество узлов и ребер  
  
for ( i = 0 ; i < N ; i++) // присваиваем элементам матрицы максимальные значения  
for ( j = 0 ; j < N ; j++) {  
mass[i][j]=max\_v;  
mass[j][i]=max\_v;  
}  
  
for (i = 0; i < M; i ++ ) // заполняем матрицу значениями из файла  
{  
fscanf(inp,"%d %d %d",&a,&b,&c);  
mass[a][b]=c;  
mass[b][a]=c;  
}  
  
fclose (inp);  
  
  
printf("Enter the first and the last point: "); // пример : " 1 3 " : 1 - начало , 3 - конец пути  
scanf ("%d %d",&start,&end);  
  
printf ("\The shortest way is: %d\n", short\_way(start,end) );  
  
getch ();  
  
return 0;  
}  
  
int short\_way (int start, int end)  
{  
/\* используем алгоритм Дейкстры \*/  
  
for ( i = 0 ; i < N ; i++) // заполняем массив, который отвечает за проход узла (если побывали в узле, то присваиваем 1, иначе 0)  
used[i]=0;  
  
for( i = 0 ; i < N ; i++)  
D[i] = mass[start][i]; // массив, который содержит кратчайший путь из заданной вершины в вершину с номером i  
  
used[start] = 1; // побывали в начальном узле  
  
for( i = 0 ; i < N-2 ; i++)  
{  
int min\_v = 1000000;  
for (j = 0; j < N; j++ )  
if (used[j]==0 && D[j]< min\_v) // если еще не побывали в вершине j, и значение в вершине с номером j меньше, чем предыдущее значение,  
{  
min\_v = D[j]; // min\_v присваивам минимальное значение.  
W = j; // W - номер узла с наименьшим значением пути  
}  
used[W]=1;  
  
for( j = 0 ; j < N ; j++)  
if (used[j]==0)  
D[j]=min( D[j], D[W] + mass[W][j]); // выбираем минимальный путь  
}  
  
if (start == end) // если начальный узел является конечным,  
D[end]=0; // то путь равен 0  
  
  
return D[end];  
}

Скрин



**Вывод:** в данной лабораторной работе я воспользовался данными с теоретической части и найденным дополнительным материалом сделал лабораторную работу а так же блок схему к выполнению процесса.