Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

по курсу «Л и ОА в ИЗ»

на тему «Поиск расстояний в графе»

**Выполнили:**

студенты группы 19ВВ2

Сидоров Н. Р.

Карамышев А. А.

Горбунов Д.А.

**Приняли:**

Юрова О. В.

Митрохин М. А.

Пенза 2020

**Название:**

Поиск расстояний во взвешенном графе

**Цель работы:**

Научиться применять алгоритм поиска расстояний во взвешенном графе на основе алгоритма поиска в ширину.

**Лабораторные задания:**

### ***Задание 1***

1. Сгенерировали (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного графа G. Вывели матрицу на экран.

2. Для сгенерированного взвешенного графа осуществили процедуру поиска

расстояний.

При реализации алгоритма в качестве очереди использовали класс queue из

стандартной библиотеки С++.

**Описание метода решения задач:**

Реализация состоит из подготовительной части, в которой все вершины

помечаются как непоcещенные. Не посещенные вершины помечаются

– 1, т.к. значение 0 и 1 могут быть расстояниями. Расстояние 0 – от исходной

вершины до самой себя.

В самой процедуре сначала создается пустая очередь, в которую

помещается исходная вершина, из которой начат обход. Расстояние до

этой вершины устанавливается равным 0 (расстояние до самой себя).

Далее итерационно, пока очередь не опустеет, из нее извлекается

первый элемент, который становится текущей вершиной. Затем в

цикле просматривается v-я строка матрицы смежности графа G(v,i). Как

только алгоритм встречает смежную с v не посещенную вершину, эта

вершина помещается в очередь и для нее обновляется вектор

расстояния. Расстояние до новой i-й вершины вычисляется как

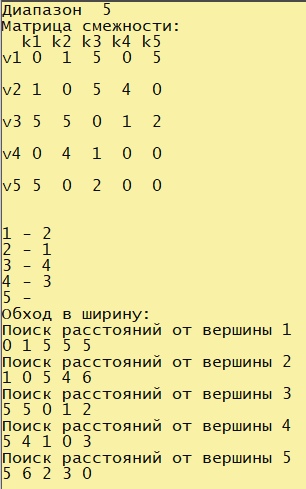
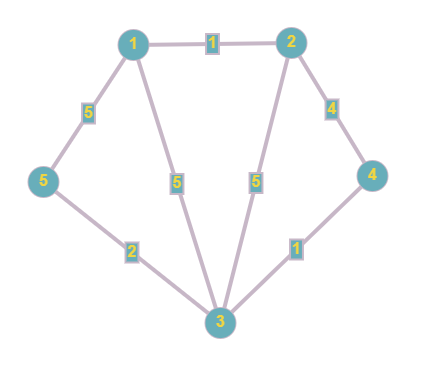
расстояние до текущей v-й вершины плюс вес ребра до новой вершины

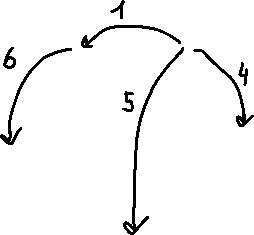
G(v,i).

После просмотра строки матрицы смежности алгоритм делает

следующую итерацию цикла или заканчивает работу, если очередь пуста.

**Результат работы программы:**



Исходя из результатов работы программы и подсчета вручную на визуализации графа можно сделать вывод, что расстояния в том и другом случае совпадают и программа работает правильно.

**Вывод:**

Научились применять алгоритм поиска расстояний во взвешенном графе на основе алгоритма обхода в ширину на примере матричного представления графа, а также визуализировать пути в виде обыкновенного графа.

Листинг 1:

#include <time.h>

#include <iostream>

#include <stack>

#include <queue>

#include <conio.h>

//работа сделана

using namespace std;

queue <int> Q;

int i, j, m, h;

int\* vis;

int\*\* graph;

void BFS() {

for (int j = 0; j < m; j++) {

printf("\nПоиск расстояний от вершины %d\n", j + 1);

for (int i = 0; i < m; i++) {

vis[i] = 1000;

}

int s = j;

Q.push(s);

int st = s;

vis[s] = 0;

while (!Q.empty())

{

s = Q.front();

Q.pop();

for (int r = 0; r < m; r++)

if ((graph[s][r] > 0) && (vis[r] > vis[s] + graph[s][r]))

{

Q.push(r);

vis[r] = vis[s] + graph[s][r];

}

}

for (int i = 0; i < m; i++)

{

printf("%d ", vis[i]);

}

}

}

void DFS\_main()

{

setlocale(LC\_ALL, "Rus");

printf("Размерность: ");

scanf\_s("%d", &m);

printf("Диапазон ");

scanf\_s("%d", &h);

h++;

graph = new int\* [m];

vis = (int\*)malloc(m \* sizeof(int));

for (int i = 0; i < m; i++) {

graph[i] = new int[m];

}

hah:

//генерация матрицы

int start;

printf("Матрица смежности: \n");

srand(time(NULL));

for (i = 0; i < m; ++i) {

for (j = i; j < m; ++j) {

graph[i][j] = graph[j][i] = rand() % h;

}

graph[i][i] = 0; //обнуление вершины

}

//нумерование столбцов

printf(" ");

for (j = 0; j < m; j++)

{

printf("k%d ", j + 1);

}

printf("\n");

//нумерование строк

for (i = 0; i < m; ++i)

{

printf("v%d ", i + 1);

vis[i] = 0;

for (j = 0; j < m; ++j)

printf("%d ", graph[i][j]);

printf("\n\n");

}

//вывод списков смежности

for (i = 0; i < m; i++)

{

printf("\n%d - ", i + 1);

for (j = 0; j < m; j++) {

if (graph[i][j] == 1) printf("%d ", j + 1);

}

}

printf("\nОбход в ширину: ");

BFS();

\_getch();

}

int Menu(void)

{

system("CLS");

int c = 0;

setlocale(LC\_ALL, "Rus");

while (c < '0' || c > '1')

{

printf(

"::=================================::\n"

"|| 1 : Обход в ширину ||\n"

"|| 0 : Выход ||\n"

"::=================================::\n"

"> ");

c = \_getch();

system("cls");

}

return c;

}

void main(void)

{

int Select;

while ((Select = Menu()) != '0' && Select != 27)

switch (Select)

{

case '1':

DFS\_main();

break;

}

}