|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №6  по курсу «Логика и основы алгоритмизации в инженерных задачах»  на тему «Поиск расстояний в графе» |
|  |
|  |
| Выполнил студент группы 19ВВ3:  Субботкин М.В.  Принял:  Митрохин М. А. |
| Пенза  2020 |

**Цель работы**

Реализовать алгоритм поиска расстояний в графе, используя алгоритм обхода в ширину.

**Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <stack>

#include <queue>

using namespace std;

int \*DIST;

void BFSD(int vertex, int f, int \*\*mass) {

queue<int>BFS;

BFS.push(vertex);

DIST[vertex] = 0;

while (!BFS.empty()) {

vertex = BFS.front();

printf("%d", vertex + 1);

BFS.pop();

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

if ((mass[vertex][i] == 1) && (DIST[i] == -1)) {

BFS.push(i);

DIST[i] = DIST[vertex] + 1;

}

}

}

}

int main() {

srand(time(NULL));

int start;

int f;

int min = 10000;

printf("Enter lenght: ");

scanf("%d", &f);

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

int\* vert = (int\*)malloc(sizeof(int) \* f);

int\*\* mass = (int\*\*)malloc(sizeof(int) \* f);

DIST = (int\*)malloc(f \* sizeof(int));

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

DIST[i] = -1;

}

printf("\n");

printf("\n");

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

vert[i] = i + 1;

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

}

printf("\n");

printf("\n");

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

mass[i] = (int\*)malloc(sizeof(int) \* f);

}

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

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

int c = rand() % 101;

if (c <= 50) {

mass[i][j] = 1;

}

else mass[i][j] = 0;

if (i == j) {

mass[i][j] = 0;

}

}

}

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

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

if (mass[i][j] != mass[j][i])

mass[i][j] = mass[j][i];

}

}

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

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

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

}

printf("\n");

}

printf("Enter start vertex :");

scanf("%d", &start);

printf("\nVisited vertexes \n");

BFSD(start-1, f, mass);

printf("\nVertexes \n");

for (int i = 1; i <= f; i++) {

printf("|%d|", i);

}

printf("\nDistance\n");

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

printf("|%d|", DIST[i]);

}

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

if ((min > DIST[i]) && (DIST[i] != 0))

min = DIST[i];

}

printf("\nShortest distance - %d , with vertex :", min);

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

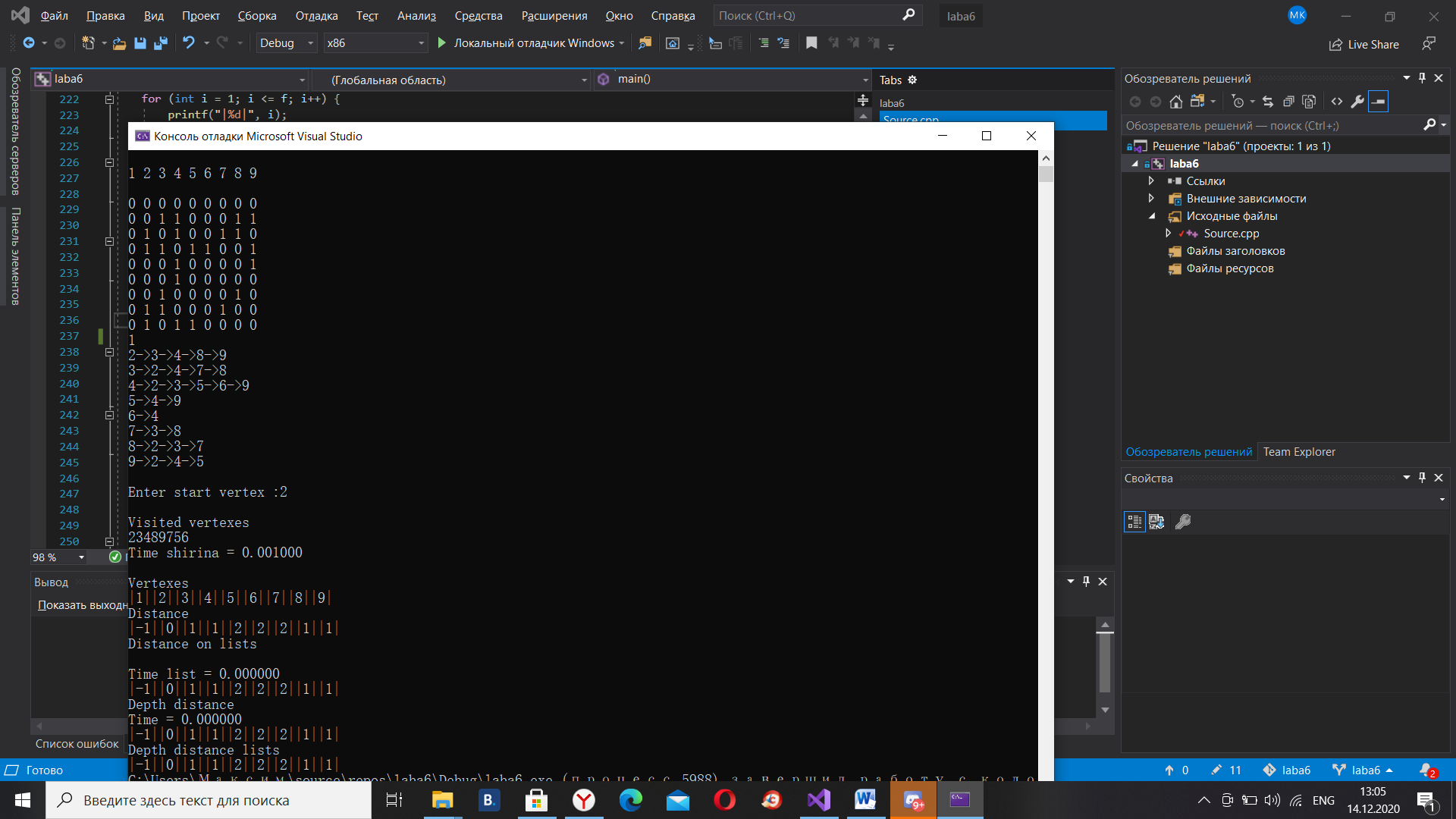
if (DIST[i] == min)

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

}

### }

### **Результат работы программы:**



**Вывод:** Реализовал алгоритм поиска расстояний в графе, используя алгоритм обхода в ширину. Обход в глубину на списках работает быстрее, чем обход на матрице, а так же сам по себе обход в глубину быстрее обхода в ширину