|  |
| --- |
| 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» |
|  |
|  |
| Выполнила студентка группы 19ВВ3:  Ханбекова Е. В.  Принял:  Митрохин М. А. |
| Пенза  2020 |

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.

 Для сгенерированного графа осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием. При  реализации алгоритма в качестве очереди используйте класс **queue** из стандартной библиотеки С++.

**Листинг:**

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <stdlib.h>

#include <queue>

#include <iostream>

using namespace std;

int BFS(int\*\*A, int\*M, int i, int N) {

queue <int> Q;

int j;

Q.push(i);

M[i] =0 ;

while (Q.empty() != true) {

i = Q.front();

Q.pop();

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

if ((A[i][j] == 1) && (M[j] ==-1)) {

M[j] = 0;

Q.push(j);

M[j]= M[i] +1 ;

}

}

}

return 0;

}

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N, i, j;

printf("Ведите размер матрицы \n");

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

srand(11);

int \*M = (int\*)malloc(N \* sizeof(int));

int \*\*A = (int \*\*)malloc(N \* sizeof(int \*));

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

A[i] = (int \*)malloc(N \* sizeof(int));

}

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

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

A[i][j] = rand() % 2;

A[j][i] = A[i][j];

if (i == j) {

A[i][j] = 0;

}

}

}

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

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

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

}

printf("\n");

}

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

M[i] = -1;

}

printf("\nОбход в ширину:\n");

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

BFS(A, M, i, N);

printf("\n");

for (int l = 0; l < N; l++) {

printf("%d ", M[l]);

}

for (int l = 0; l < N; l++) {

M[l] = -1;

}

}

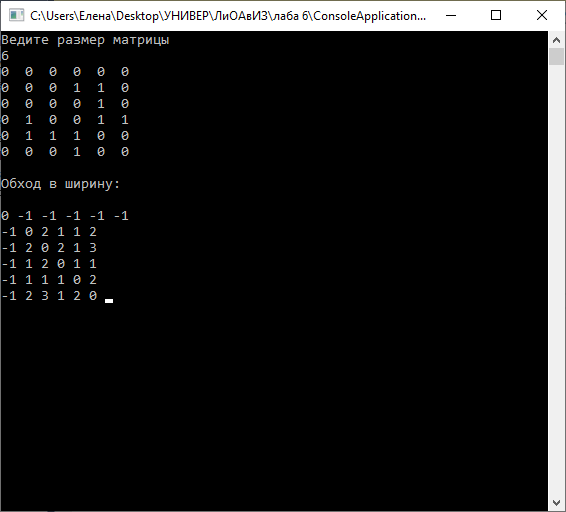
free(A);

free(M);

\_getch();

}

**Результат работы программы:**



**Вывод:**

В результате работы был реализован поиск путей в графе на основе обхода графа в глубину. Обход графа в ширину был изменен путем увеличения расстояния на 1 для вершин каждого следующего уровня глубины.