Министерство высшего образования и науки Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Отчет

По лабораторной работе №8

По курсу «Логика и основы алгоритмизации в инженерных задачах»

На тему «Обход графа в ширину»

Работу выполнили студенты группы 21ВВ2:

Хабибулин А.М.

Щеглов Д.А.

Приняли:

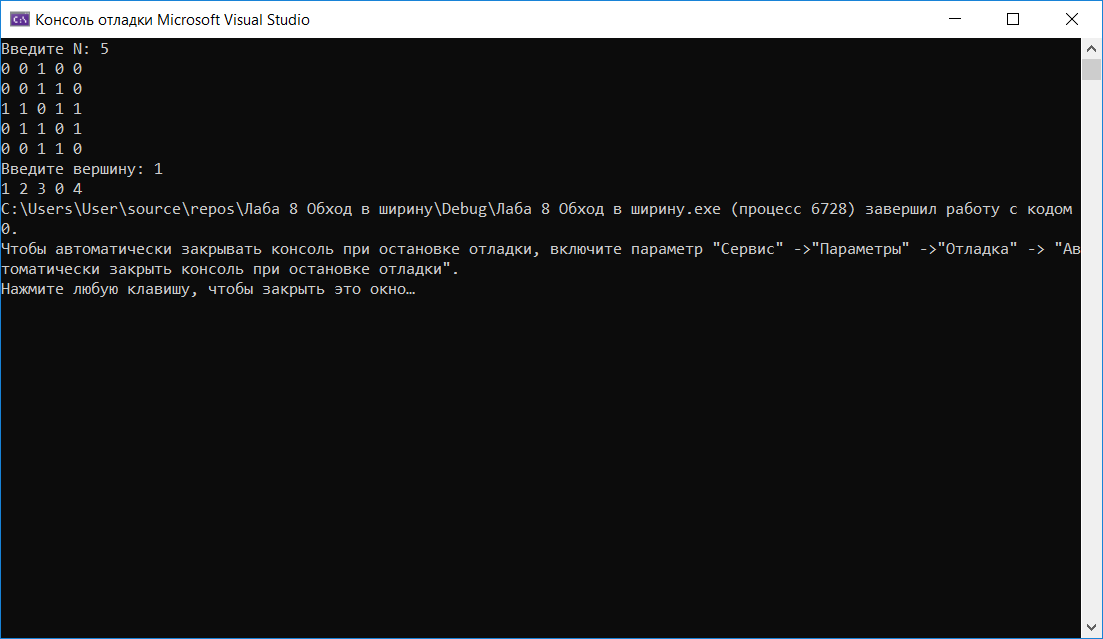
Юрова О.В.

Митрохин М.А.

**Ход работы:**

Сгенерировали с помощью генератора случайных чисел матрицу  
смежности для неориентированного графа. Матрица выводится на экран.  
 Осуществили процедуру обхода графа в ширину. Данный обход заключается в посещении вершин по уровням удаленности от исходной вершины. Уровень за уровнем посещаются все вершины, посещенные вершины помечаются. Для обхода всех уровней по порядку используется очередь. При реализации алгоритма в качестве очереди использовался класс queue из стандартной библиотеки С++. Просматривая строку матрицы смежности для текущей вершины, все не посещённые вершины по соседству с ней помещаются в очередь. На следующей итерации текущей вершиной становится та, которая стоит в очереди первой и уже её не посещённые соседи будут помещены в очередь после тех вершин, которые были помещены туда ранее.

**Результат работы программы показан на рис.1**

  
**Рис.1 -результат работы программ**

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <queue>

#include <locale.h>

#include <time.h>

using namespace std;

void BFS(int\*\* G, int\* visited, int num, int N) {

queue <int> q;

int i;

q.push(num); //помещаем вершину исходную в очередь

visited[num] = 1; // отмечаем вершину как посещенную

while (!q.empty()) { // пока очередь не пуста, выполняем

num = q.front(); // устанавливаем текущую вершину

q.pop(); // удаляем первый элемент из очереди

printf("%d ", num);

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

if (visited[i] == 0 && G[num][i] == 1) {

q.push(i); // помещаем в очередь i

visited[i] = 1; // отмечаем вершину как посещенную

}

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

int\*\* a;

int N;

printf("Введите N: ");

scanf("%d", &N);

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

//mass = (int\*)malloc(N \* sizeof(int));

srand(time(NULL));

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

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

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

a[i][j] = 0;

}

}

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

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

if (i != j) {

int num = rand() % 2;

a[i][j] = num;

a[j][i] = num;

}

}

}

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

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

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

}

printf("\n");

}

int len = N;

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

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

visited[i] = 0;

}

printf("Введите вершину: ");

scanf("%d", &N);

BFS(a, visited, N, len);

free(a);

free(visited);

return 0;

}

**Вывод:**

В ходе работы была написана программа, которая генерирует неориентированный граф и осуществляет его обход в ширину. Задачей данного обхода является прохождение всех вершин графа по определенному порядку. Данный алгоритм можно применить для поиска пути, нахождения решения какой-либо задачи с наименьшим числом ходов и т.д.