Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №8

по теме «Обход графа в ширину»

Выполнили:

Студенты группы 21ВВ2

Козлова К.С.

Кривенкова В.С.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза 2022

**Цель работы:**

Изучить основные алгоритмы обхода графа и научиться решать задачи обхода графа на основе поиска в ширину.

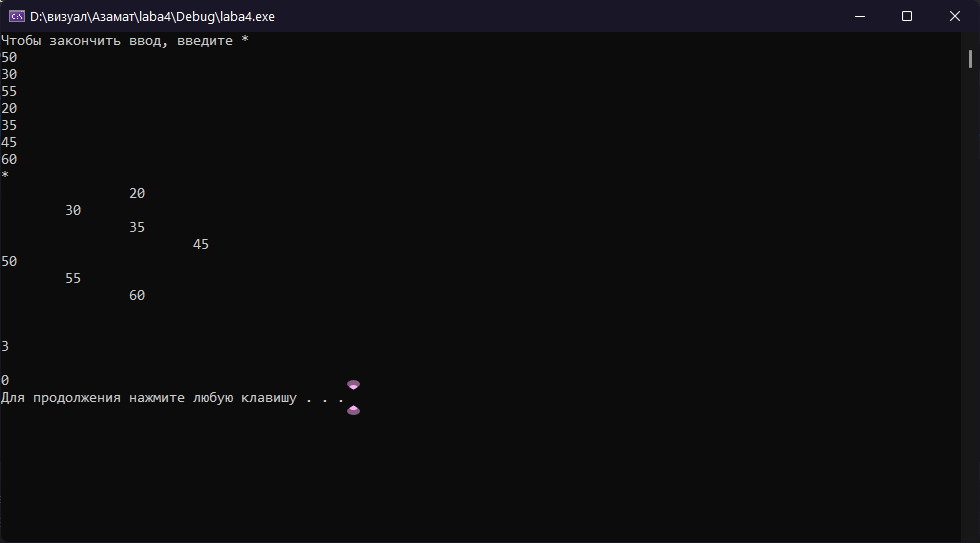
**Лабораторные работы:**

Задание 1

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

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

**Результат работы программы:**



**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<list>

#include <deque>

#include<queue>

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <locale.h>

using namespace std;

void BFS(int s);

int\* visited;

int\*\* a;

int n;

int main()

{

srand(time(0));

int s;

setlocale(LC\_ALL, "rus");

printf("Введите вершину с которой начинается обход:");

scanf("%d", &s);

printf("Введите количество элементов для графа:");

scanf("%d", &n);

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

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

{

visited[i] = 0;

}

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

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

{

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

}

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

{

for (int j = i; j < n; j++)

{

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

if (i == j)

{

a[i][j] = 0;

}

a[j][i] = a[i][j];

}

}

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

{

for (int j = 0; j < n; j++)

{

printf("%-5d ", a[i][j]);

}

printf("\n");

}

printf("\n");

printf("\n");

printf("\n");

printf("\n");

BFS(s);

printf("\n");

printf("\n");

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

{

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

}

}

void BFS(int s)

{

queue<int> S;

S.push(s);

visited[s]++;

while (!S.empty())

{

s = S.front();

S.pop();

cout << s << " ";

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

{

if (!visited[i] && a[s][i])

{

S.push(i);

visited[i]++;

}

}

}

}

**Вывод:** в ходе данной лабораторной работы мы научились реализовывать основные алгоритмы обхода графа и научились решать задачи обхода графа на основе поиска в ширину.