Пензенский Государственный университет

Кафедра «Вычислительной техники»

Отчёт

По лабораторной работе №5

По дисциплине «Л и ОА в ИЗ»

На тему: «Обход графа в ширину»

***Выполнил студент группы 19ВВ1:***

Даврешян А.

**Приняли:**

Митрохин М.А.

Юрова О.В.

Пенза 2020

**Цель работы:** Изучить алгоритм обхода в ширину.

**Задание:**

1. Сгенерировать (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществить процедуру обхода в ширину, реализованную в соответствии с приведенным выше описанием. При  реализации алгоритма в качестве очереди используйте класс **queue** из стандартной библиотеки С++.

**Ход работы**

Обход графа в ширину – еще один распространенный способ обхода графов.

Основная идея такого обхода состоит в том, чтобы посещать вершины по уровням удаленности от исходной вершины. Удалённость в данном случае понимается как количество ребер, по которым необходимо перейти до достижения вершины.

**Листинг**

// laba5.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include "iostream"

#include "conio.h"

#include "locale.h"

#include <queue>

using namespace std;

int \*vis; // указатель на вектор

int \*\*matr;//указатель для массива указателей

void BFS(int s , int n)

{

queue <int> Q;

Q.push(s);

vis[s] = 1;

while(!Q.empty())

{

s = Q.front();

printf("%d", s);

Q.pop();

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

{

if((matr[s][i] == 1) && !vis[i])

{

Q.push(i);

vis[i] = 1;

}

}

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int n,i,j,x, Nach;

cout << "Input matrix order:";

cin >> n;

cout << "Input value vis:";

cin >> x;

vis = new int[x];

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

//выделение памяти под массив указателей

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

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

}

//выделение памяти для массива значений

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

{

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

{

if(i == j)

{

matr[i][j] = 0;

}

if (i < j)

{

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

matr[j][i] = matr[i][j];

}

}

}

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

{

vis[i] = 0;

}

printf( "Result: ");

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

{

cout << "\n";

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

printf("%d\t",matr[i][j]);

}

printf( " \n ");

printf( " \n ");

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

{

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

}

printf( " \n ");

printf("Vvedite vershinu nachala obhoda:\n");

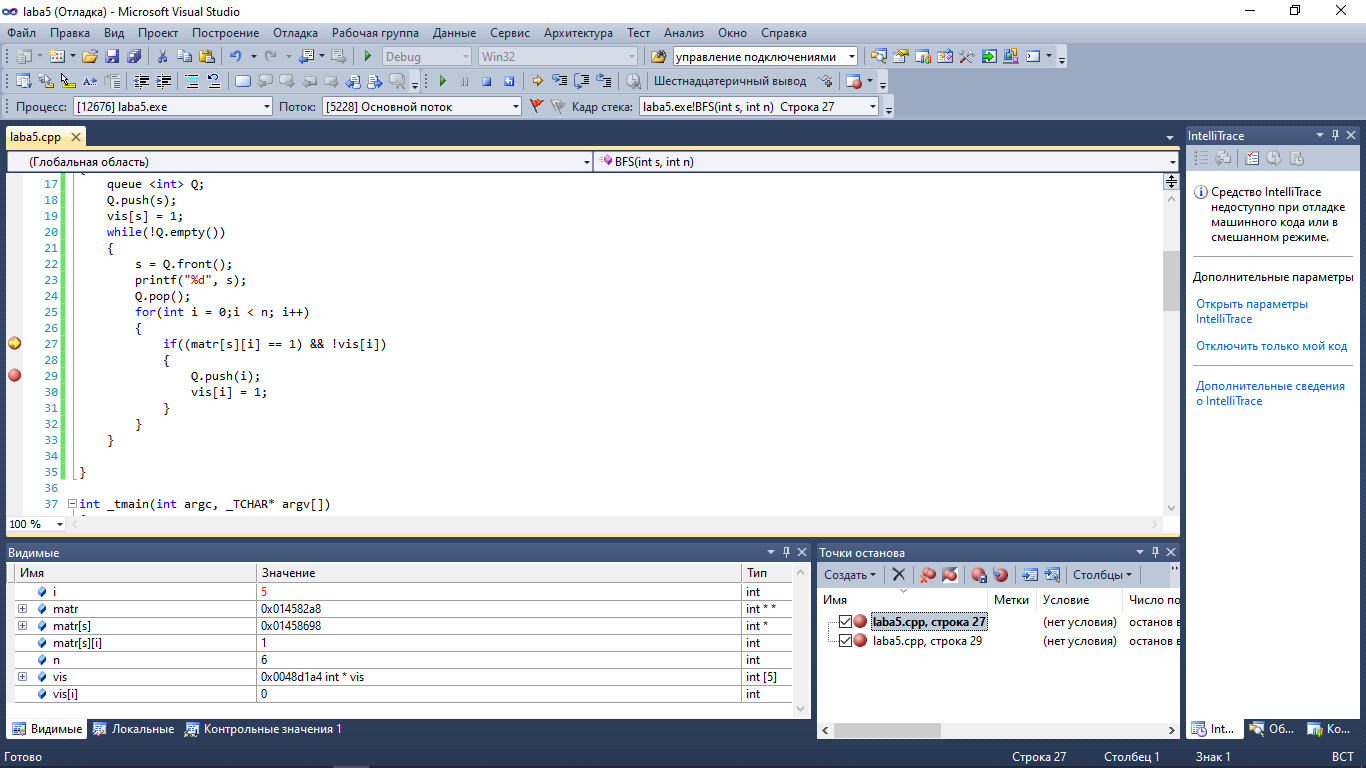
scanf("%d", &Nach);

BFS(Nach, n);

getch();

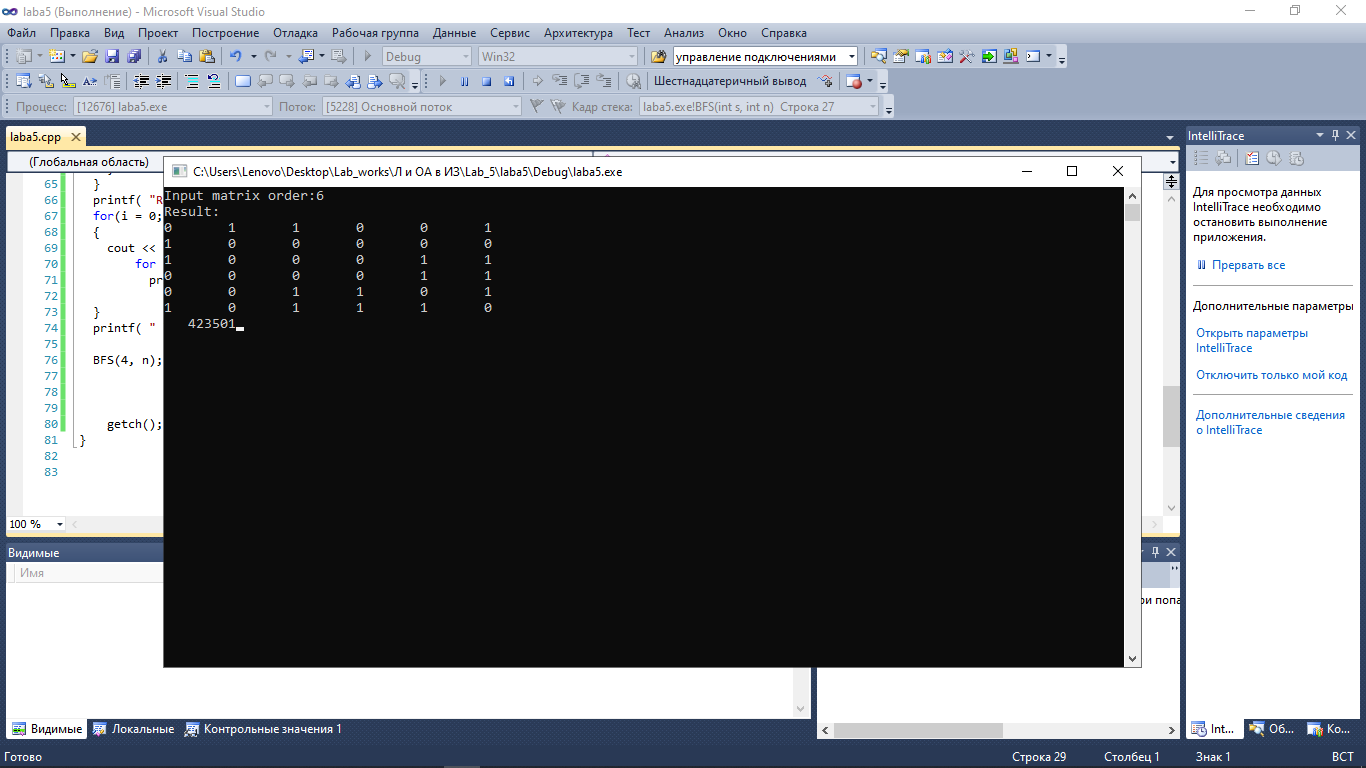
}

**Результат трассировки программы**

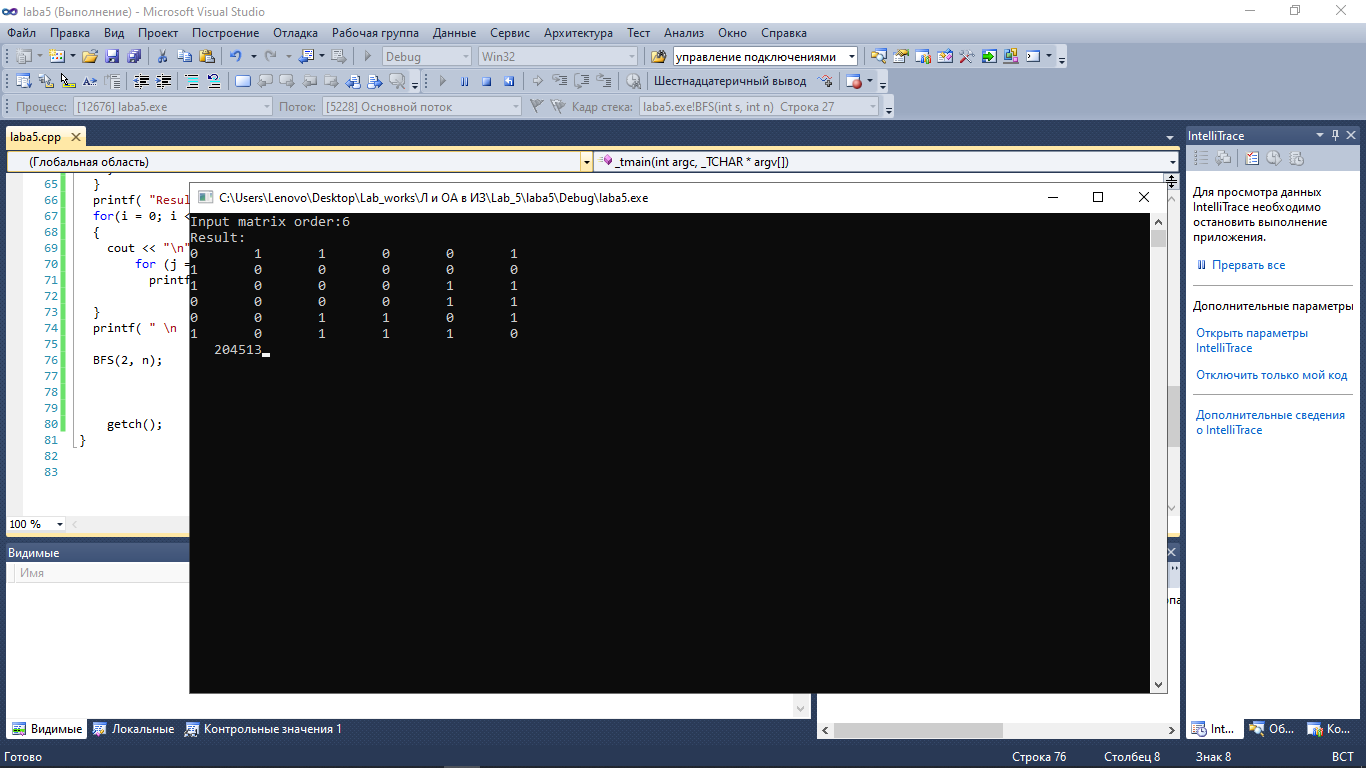


**Результат работы программы**

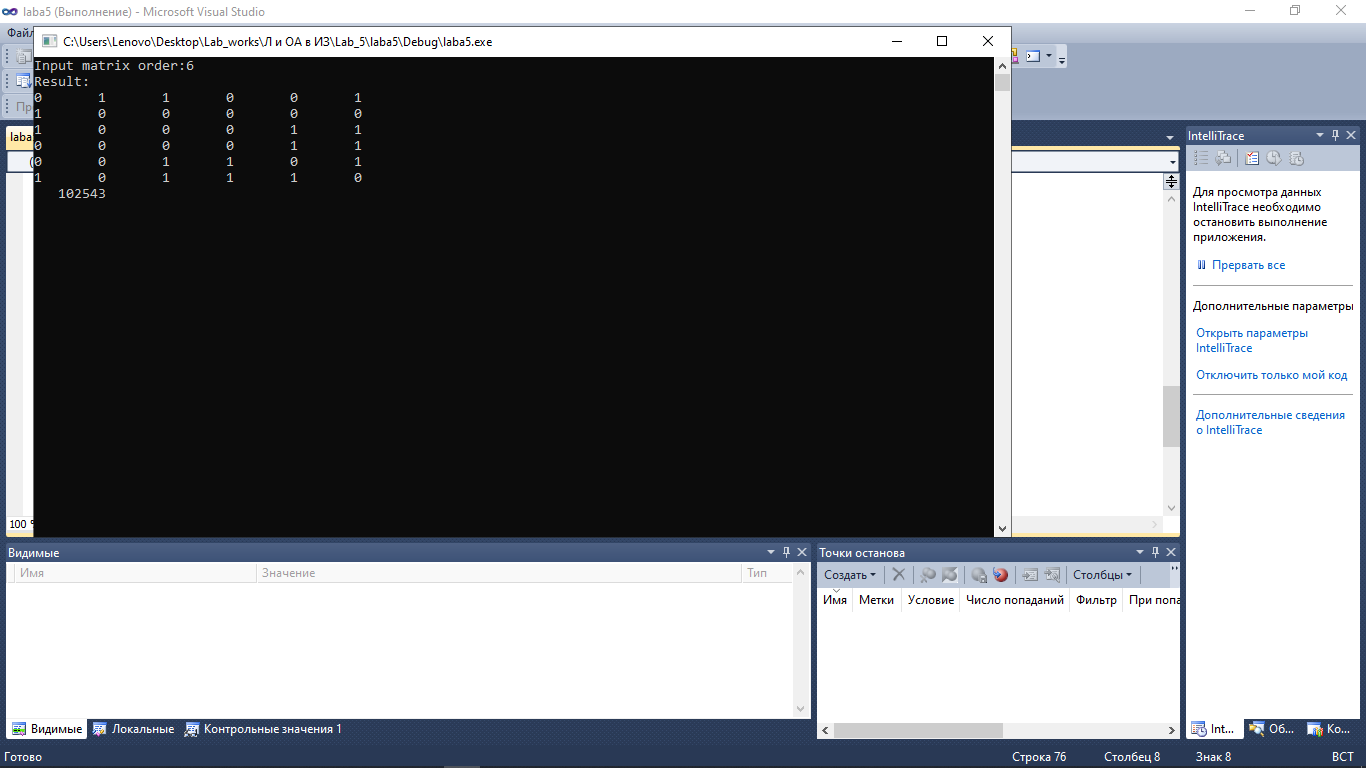
Результат обхода матрицы 6х6 с вершины 4



Результат обхода матрицы 6х6 с вершины 2



Результат обхода матрицы 6х6 с вершины 1



**Вывод:** В ходе данной работы был изучен алгоритм обхода в ширину. Так же данный алгоритм был реализован для матрицы , размер которой пользователь может вводить с экрана.