Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

К курсовому проектированию

По курсу «Логика и основы алгоритмизации

в инженерных задачах»

На тему «Реализация алгоритма нахождения изолированных вершин в графе»

Выполнил:

Студент группы 20ВВ4:

Исаев С. Д.

Приняли:

Митрохин М.А.

Юрова О.В.

Пенза 2020

**Содержание**

[Введение 7](#_Toc61474523)

[1. Постановка задачи 9](#_Toc61474524)

[2. Теоретическая часть задания 10](#_Toc61474525)

[3. Описание алгоритма программы 12](#_Toc61474526)

[4. Описание программы 14](#_Toc61474527)

[5. Тестирование 17](#_Toc61474528)

[Заключение 24](#_Toc61474529)

[Список литературы 25](#_Toc61474530)

[Приложение. 26](#_Toc61474531)

# Введение

Язык программирования Cи был создан Деннисом Ритчи из AT&T Bell Laboratories как язык программирования высокого уровня для разработки операционной системы UNIX. В настоящее время язык широко используется в различных областях. Cи особенно популярен у системных программистов, потому что позволяет писать программы просто и кратко.

Язык Си считается языком системного программирования, хотя он удобен и для написания прикладных программ. Среди преимуществ языка Си следует отметить переносимость программ на компьютеры различной архитектуры и из одной операционной системы в другую, лаконичность записи алгоритмов, логическую стройность программ, а также возможность получить программный код, сравнимый по скорости выполнения с программами, написанными на языке ассемблера.

**Теория графов** - один из обширнейших разделов дискретной математики, широко применяется в решении экономических и управленческих задач, в программировании, химии, конструировании и изучении электрических цепей, коммуникации, психологии, психологии, социологии, лингвистике, других областях знаний. **Теория графов** систематически и последовательно изучает свойства графов, о которых можно сказать, что они состоят из множеств точек и множеств линий, отображающих связи между этими точками.

Основателем теории графов считается Леонард Эйлер (1707-1882), решивший в 1736 году известную в то время задачу о кёнигсбергских мостах.

**Графом называется** система объектов произвольной природы (вершин) и связок (рёбер), соединяющих некоторые пары этих объектов.

В теории графов вершиной называется фундаментальная единица, образующая графы.

Степенью вершины графа называется число рёбер, инцидентных ей. Вершина называется изолированной, если её степень равна нулю. То есть это вершина, не являющаяся конечной ни для какого ребра.

В качестве среды разработки мною была выбрана среда Microsoft Visual Studio 2019, а язык программирования – С.

# Постановка задачи

Цель исследования - 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 прейти до

достижения вершины.

# Теоретическая часть задания

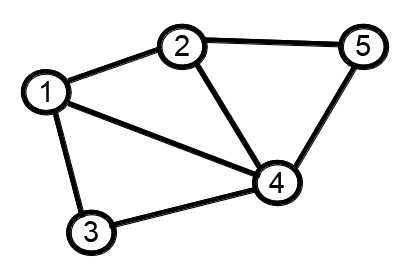
**Графом** называется геометрическая фигура, состоящая из точек и соединяющих их линий. Точки называются **вершинами** графа, а линии — **ребрами**. Каждому ребру сопоставлены две вершины — концы ребра. Пример графа представлен на рис.1.

Рисунок 1. Пример графа

Если некоторое ребро u соединяет две вершины A и B графа, то говорят, что ребро u **инцидентно** вершинам A и B, а вершины в свою очередь **инцидентны** ребру u. Вершины, соединенные ребром, также называют **смежными**.

**Степенью вершины** называют количество ребер, для которых она является концевой (при этом петли считают дважды).

# Описание алгоритма программы

Для реализации программы был применен данный алгоритм:

Пусть задан невзвешенный ориентированный граф G=(V,E)G=(V,E), в котором выделена исходная вершина ss. Требуется найти длину кратчайшего пути (если таковой имеется) от одной заданной вершины до другой. Частным случаем указанного графа является невзвешенный неориентированный граф, т.е. граф, в котором для каждого ребра найдется обратное, соединяющее те же вершины в другом направлении.

Для алгоритма нам потребуются очередь и множество посещенных вершин waswas, которые изначально содержат одну вершину ss. На каждом шагу алгоритм берет из начала очереди вершину vv и добавляет все непосещенные смежные с vv вершины в waswas и в конец очереди. Если очередь пуста, то алгоритм завершает работу.

Ниже приведен пример алгоритма обхода графа в ширину.

void DFS(int v, int\*\* matrix, int size,int\* NUM) {

NUM[v] = 1;

printf("%d", v);

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

if (matrix[v][i] == 1 && NUM[i] == 0)

{

DFS(i, matrix, size, NUM);

}

}

}

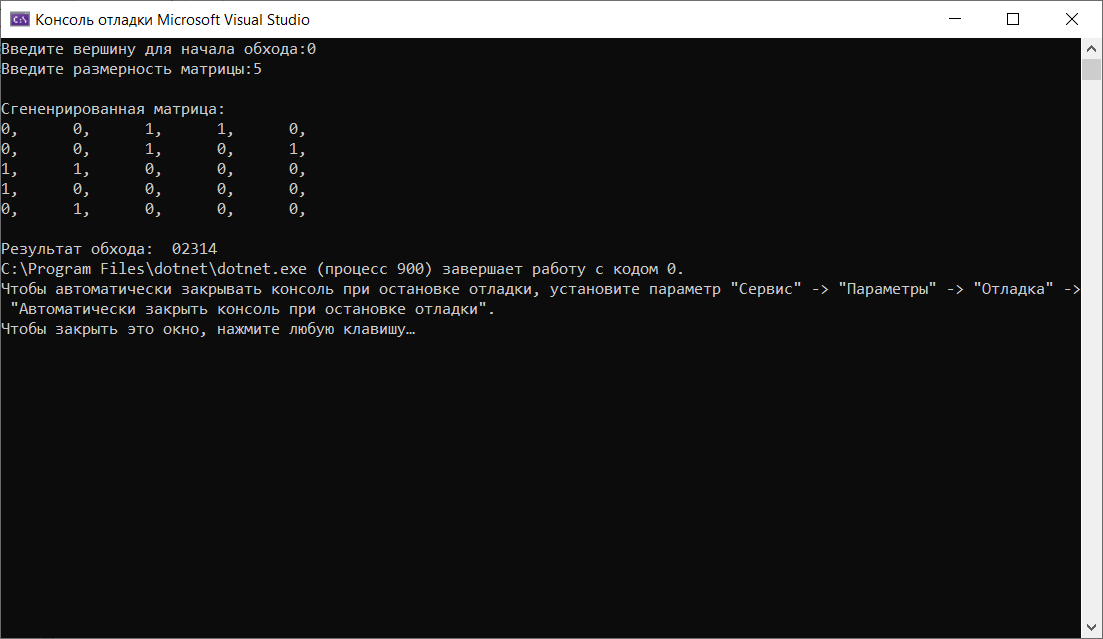
Полный код программы можно увидеть в Приложении А.

# Описание программы

Проект был создан в виде консольного приложения (Visual C++).

Пользователю предлагается выбрать количество вершин графа, и с какой из них начать обход.

После ввода данных появляется матрица смежности и сам обход.



# Тестирование

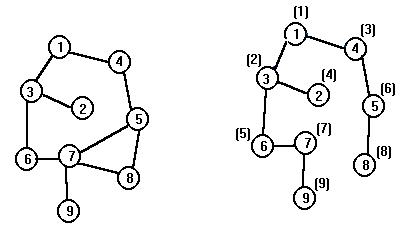
Таблица 1 – Описание поведения программы при тестировании

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Ожидаемый результат** | **Полученный результат** |
| Запуск программы | Открытие консольного приложения | Верно |
| Ввод вершины для обхода | Результат в зависимости от выбора пользователя | Верно |
| Ввод размера матрицы смежности | Вывод матрицы смежности | Верно |
| Вывод последовательности обхода в ширину | Вывод последовательности обхода в ширину | Верно |

В результате тестирования было доказано, что алгоритм работает верно.

1. **Ручной расчёт задачи**

Проведем ручную проверку задачи



Результат: 134265789

Сравнив результат работы программы с ручным расчетом, можно сказать, что программа работает верно

# Заключение

Таким образом, в процессе создания данного проекта разработана программа, реализующая алгоритм обхода графа в ширину в среде разработки Microsoft Visual Studio 2019.

При выполнении данной работы были получены навыки работы с графом, представленным матрицей смежности и был разработан и реализован алгоритм обхода графа в ширину . Так же были улучшены навыки программирования на языке Си.

Программа имеет небольшой, но достаточный для использования функционал возможностей и интерфейс.

# Список литературы

1. Теория графов. Алексеев В.Е., Захарова Д.В. Электронное учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 57 с.
2. Зыков А.А. Основы теории графов. - М.:Наука, 1987, 384 с.
3. Мельников О. И. Незнайка в стране графов: Пособие для учащихся. Изд. 3-е, стереотипное. М.: КомКнига, 2007. — 160 с.
4. Брайан Керниган, Деннис Ритчи, Алан Фьюэр. Язык программирования Си. Задачи по языку Си. — Москва: Финансы и статистика, 1985. — 279 с.
5. Пахомов Б.И. C/C++ и MS Visual C++ 2008 для начинающих. Учебник. 2009 год. 609 стр.
6. Т. А. Павловская. C/C++. Программирование на языке высокого уровня. Из серии "300 лучших учебников".2003 год. 461 стр.

## Приложение.

**Листинг программы.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <locale>

#include <stdio.h>

#include <conio.h>

#include <locale.h>

#include <stdlib.h>

#include <malloc.h>

#include <iostream>

void DFS(int v, int\*\* matrix, int size,int\* NUM) {

NUM[v] = 1;

printf("%d", v);

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

if (matrix[v][i] == 1 && NUM[i] == 0)

{

DFS(i, matrix, size, NUM);

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

int v, n;

int\* NUM;

printf("Введите размерность матрицы: ");

scanf("%d", &n);

int\*\* M;

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

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

{

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

}

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

printf("Сгенерированные матрицы:");

printf("\n");

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

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

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

printf("%3d,", M[i][j]);

}

printf("\n");

}

printf("\n\n");

printf("Введите вершину, с которой будет начинаться обход: ");

scanf("%d", &v);

printf("\n");

printf("Результат: ");

while (NUM[v] != 1)

{

DFS(v, M, n, NUM);

}

getchar(); getchar();

}