МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра алгебры, геометрии и дискретной математики**

Направление подготовки «Прикладная математика и информатика»

**ОТЧЕТ**

по учебной практике

на тему:

**«Изучение алгоритма поиска в ширину и задачи нахождения кратчайшего пути»**

**Выполнил:** студент группы 381603-1

**\_\_\_\_\_\_\_\_\_\_\_\_\_** Шахнов А. В.

Подпись

**Научный руководитель:**

Доцент кафедры АГДМ, кандидат физико-математических наук

**\_\_\_\_\_\_\_\_\_\_\_\_\_** Сорочан С.В.

Подпись

Нижний Новгород

2019

**Содержание**

1. **Введение**
2. **Постановка учебно-практической задачи**
3. **Руководство пользователя**
4. **Наблюдения и выводы**

# Введение

Графы являются существенным элементом математических моделей в самых разнообразных областях науки и практики. Они помогают наглядно представить взаимоотношения между объектами или событиями в сложных системах. Многие алгоритмические задачи дискретной математики могут быть сформулированы как задачи, так или иначе связанные с графами, например, задачи, в которых требуется выяснить какие-либо особенности устройства графа, или найти в графе часть, удовлетворяющую некоторым требованиям, или построить граф с заданными свойствами.

# Постановка учебно-практической задачи

# Основные понятия и определения

*Обыкновенным графом* называется пара G = (V, E), где V – конечное множество, E – множество неупорядоченных пар различных элементов из V. Элементы множества V называются *вершинами графа*, элементы множества E – его *ребрами*.

Если ребра имеют направления – граф называют *ориентированным*. Если между любой парой вершин существует как минимум один путь – граф называют *связным*. Разные рёбра могут иметь одинаковые начала и концы. В таком случае говорят, что графе допускаются *кратные рёбра*, а граф называют *мультиграфом*. Ребро, соединяющее вершину с самой собой называют *петлёй*. Если такие ребра не допускаются – говорят, что рассматривается граф без петель.

Также есть несколько разных *способов задания* графов.

Пусть G - граф с n вершинами, причем VG = {1,2…n}. Построим квадратную матрицу A порядка n, в которой элемент Aij, стоящий на пересечении строки с номером i и столбца с номером j, определяется следующим образом: Aij = 1, если между i и j вершинами существует ребро, и Aij = 0, если ребро отсутствует.

Матрица A называется *матрицей смежности* графа. Матрицу смежности можно построить и для ориентированного графа, и для неориентированного, и для графа с петлями. Для обыкновенного графа она обладает двумя особенностями: из-за отсутствия петель на главной диагонали стоят нули, а так как граф неориентированный, то матрица симметрична относительно главной диагонали. Обратно, каждой квадратной матрице порядка n, составленной из нулей и единиц и обладающей двумя указанными свойствами, соответствует обыкновенный граф с множеством вершин {1,2…n}.

Другая матрица, ассоциированная с графом - это *матрица инцидентности.* Для ее построения занумеруем вершины графа числами от 1 до n, а ребра - числами от 1 до m. Матрица инцидентности I имеет n строк и m столбцов, а ее элемент Iij равен 1, если вершина с номером i инцидентна ребру с номером j, в противном случае он равен нулю.

В практической работе используются связные неориентированные графы без петель.

# Поиск в ширину Поиск в ширину – (англ. *breadth-first search*, BFS) – один из методов обхода графа.

Идея поиска в ширину состоит в том, чтобы посещать вершины в порядке их удаленности от некоторой заранее выбранной или указанной стартовой вершины a. Иначе говоря, сначала посещается сама вершина a, затем все вершины, смежные с a, то есть находящиеся от нее на расстоянии 1, затем вершины, находящиеся от a на расстоянии 2, и т.д.

Основная особенность поиска в ширину, отличающая его от других способов обхода графов, состоит в том, что в качестве активной вершины выбирается та из открытых, которая была посещена раньше других. Именно этим обеспечивается главное свойство поиска в ширину: чем ближе вершина к старту, тем раньше она будет посещена. Для реализации такого правила выбора активной вершины удобно использовать для хранения множества открытых вершин очередь – когда новая вершина становится открытой, она добавляется в конец очереди, а активная выбирается в ее начале.

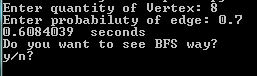
**Поиск кратчайшего пути**Задача о кратчайшем пути является одной из важнейших задач теории графов. Значимость данной задачи определяется ее различными практическими применениями. Например, в GPS-навигаторах осуществляется поиск кратчайшего пути между двумя перекрёстками. В качестве вершин выступают перекрёстки, а дороги являются рёбрами, которые лежат между ними. Если сумма длин дорог между перекрёстками минимальна, тогда найденный путь самый короткий. Задача поиска кратчайшего пути на графе может быть определена для неориентированного, ориентированного или смешанного графа. В моей практической работе рассмотрена задача для неориентированного графа. Для смешанного и ориентированного графа дополнительно должны учитываться направления ребер.

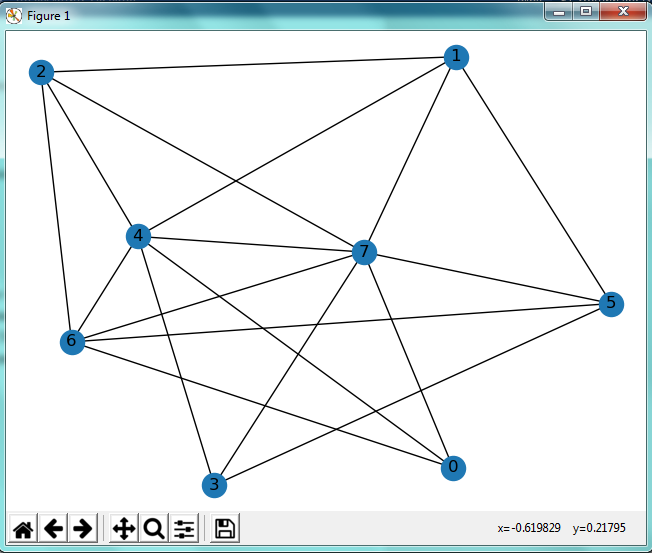
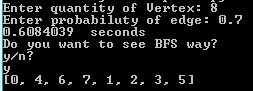
# Руководство пользователя

Для начала работы программы необходимо запустить проект Graph.py.  
Первым делом программа предложит ввести количество вершин в будущем графе.   

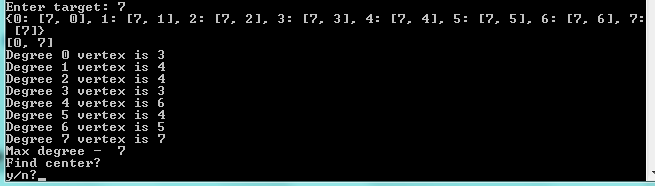

Далее вручную вводится вероятность существования ребра между вершинами.



После ввода вероятности случайным образом генерируется случайный обыкновенный граф (не всегда связный)  
   
На экран будет выведено время, за которое был осуществлён поиск, а так же предложение вывести результат поиска в ширину. Для наглядности не рекомендуется выводить результат при количестве вершин больше 50. Также, если количество вершин меньше или равно 15, программа нарисует граф, сгенерированный ранее, дав возможность проверить результаты поиска. Изображение будет сохранено в папке с проектом. Так же стоит добавить, что поиск ведётся от вершины с номером “0”



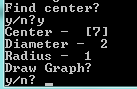
После закрытия окна с рисунком графа будет осуществлён поиск кратчайшего пути от нулевой вершины до введённой пользователем (так же присутствует реализация поиска не от нулевой вершины, а от заданной вручную).



Программа вернёт результат в виде списка, элементами которого являются вершины. Так же выполняется поиск путей от введённой вершины до всех остальных, результат будет выдан в формате словаря, где перед двоеточием стоит номер вершины, до которой ищется путь, а после него список пройденных вершин.

В программе реализовано вычисление степеней вершин графа, в конце пользователю выводится максимальная из них.

После вывода степеней пользователю будет предложено найти центр, радиус и диаметр графа.



А затем предложение снова (или же впервые) нарисовать граф, чтобы проверить результат.

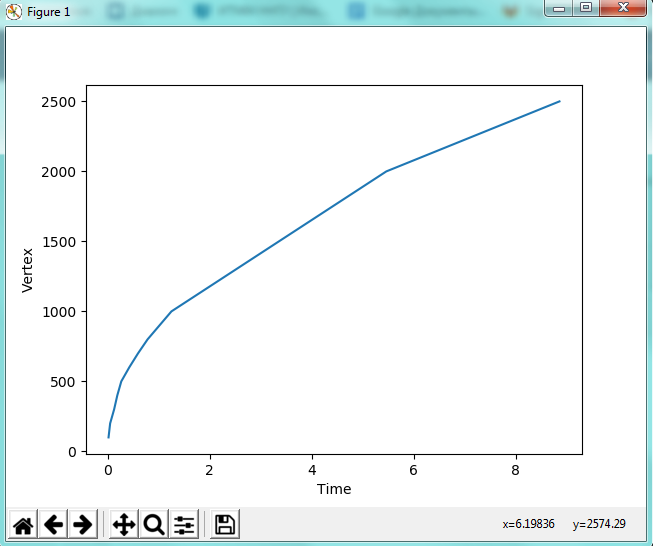
В конце пользователю будет предложено заново ввести количество вершин и произвести те же вычисления уже в другом графе.

**Наблюдения и выводы**

В данной работе были разработаны методы поиска в ширину, нахождения кратчайшего пути в графе, поиск центра, радиуса и диаметра графа, а также вычисление степеней вершин.

Интересное наблюдение – граф генерируется таким образом, что максимальное расстояние между любыми вершинами обычно бывает не больше 5. Даже в графах с большим количеством вершин (>=1000) можно попасть в любую вершину из первой за 3-4 перехода.

В рамках учебной практики был проведён следующий эксперимент:  
Замерялось среднее время работы 10 поисков в ширину на графах из 100,200,300…1000,2000,2500 вершин, и строился график зависимости времени от числа вершин.

Кривая, полученная в результате эксперимента, имеет параболический вид.

# 