

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №5**  
**Вариант №1**

Выполнили:

Смирнов И.И.

Касьяненко В.М.

Проверил:

Мусаев А.А.

Санкт-Петербург

2022

## СОДЕРЖАНИЕ

|                          |   |
|--------------------------|---|
| Введение.....            | 3 |
| 1 Задание 1 .....        | 4 |
| 1.1 Поиск в глубину..... | 4 |
| 1.2 Поиск в ширину.....  | 5 |
| Задание 2 .....          | 6 |
| 2.1 Лабиринт .....       | 6 |
| Заключение .....         | 7 |
| Список литературы .....  | 8 |
| Приложение .....         | 9 |

## ВВЕДЕНИЕ

Для становления хорошим специалистом в области программирования на языке Python необходимо знать основные алгоритмы и функционал языка.

Цель данной работы – ознакомление с алгоритмами поиска в глубину и ширину на языке программирования Python.

В ходе выполнения лабораторной работы были решены следующие задачи:

- создание программ с алгоритмом поиска в глубину и ширину;
- создание программы для прохождения лабиринта.

Задания, которые необходимо выполнить:

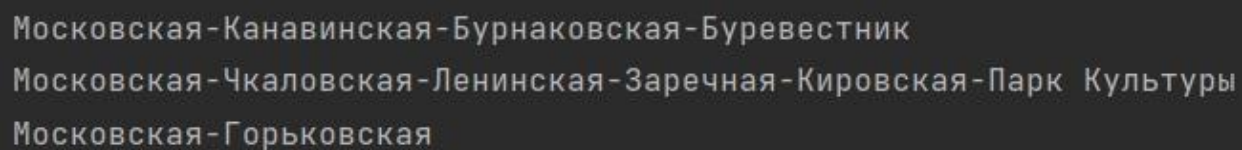
1. Задание 1: придумать и решить задачу для алгоритма поиска в глубину и ширину. Объяснить, почему для решения поставленных задач были выбраны именно эти алгоритмы поиска.

2. Задание 2: написать алгоритм, который найдет выход из лабиринта, если дана случайная квадратная матрица, заполненная нулями и единицами, где предполагается, что 0 – это проход, а 1 – это стена.

## 1 ЗАДАНИЕ 1

### 1.1 Поиск в глубину

Для поиска в глубину была придумана задача, где необходимо найти сколько существует таких конечных станций метрополитена Нижнего Новгорода, до которых мы можем добраться от станции «Московская». Алгоритм поиска в глубину подходит здесь больше, чем остальные, так как он позволяет найти все пути до конечных точек в быстрые сроки (рисунок 1).



```
Московская-Канавинская-Бурнаковская-Буревестник  
Московская-Чкаловская-Ленинская-Заречная-Кировская-Парк Культуры  
Московская-Горьковская
```

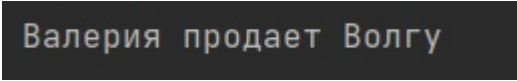
Рисунок 1 – Вывод конечных точек

## 1.2 Поиск в ширину

Для алгоритма поиска в ширину была придумана задача. Допустим, мы хотим купить машину у наших друзей или друзей друзей. Мы знаем, что у каждого человека первая буква имени соответствует первой букве названия марки автомобиля. Нам нужен человек с Волгой, которого мы ближе всего знаем по связям.

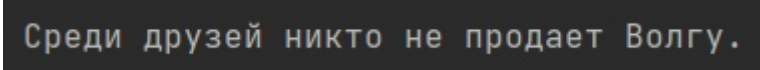
Для решения этой задачи был использован алгоритм поиска в ширину, так как он подходит больше, чем остальные алгоритмы, для нахождения кратчайшей связи с необходимым человеком.

Если человек нашелся, то выводится на экран его имя (рисунок 2), а если, то программа выдает ошибку (рисунок 3).



```
Валерия продает Волгу
```

Рисунок 2 – Вывод программы, если человек нашелся



```
Среди друзей никто не продает Волгу.
```

Рисунок 3 – Вывод программы, если никого не нашлось

## ЗАДАНИЕ 2

### 2.1 Лабиринт

Для решения задачи с лабиринтом был выбран алгоритм поиска в ширину, так как он позволяет наиболее эффективно и быстро построить маршрут для выхода из лабиринта. Для начала создается нулевая матрица, в которую будет записываться сумма всех пройденных клеток, по которой в конце будет выбран кратчайший маршрут (рисунок 4) и его координаты (рисунок 5).

|   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0 | 8 | 9 | 10 | 11 | 0  | 21 | 0  | 23 | 24 | 25 | 26 | 0  | 0  | 0  |
| 0 | 7 | 0 | 0  | 0  | 0  | 20 | 0  | 22 | 0  | 0  | 25 | 0  | 0  | 0  |
| 0 | 6 | 0 | 22 | 21 | 20 | 19 | 0  | 21 | 0  | 0  | 24 | 0  | 0  | 0  |
| 0 | 5 | 0 | 0  | 0  | 0  | 18 | 0  | 20 | 21 | 0  | 23 | 0  | 0  | 0  |
| 0 | 4 | 5 | 6  | 7  | 0  | 17 | 0  | 19 | 0  | 0  | 22 | 0  | 0  | 0  |
| 0 | 3 | 0 | 0  | 0  | 0  | 16 | 0  | 18 | 0  | 0  | 21 | 0  | 27 | 0  |
| 1 | 2 | 0 | 0  | 15 | 0  | 15 | 0  | 17 | 0  | 21 | 20 | 0  | 26 | 27 |
| 0 | 3 | 0 | 15 | 14 | 13 | 14 | 15 | 16 | 0  | 0  | 19 | 0  | 25 | 0  |
| 0 | 4 | 0 | 0  | 0  | 12 | 0  | 0  | 17 | 18 | 0  | 18 | 0  | 24 | 0  |
| 0 | 5 | 6 | 0  | 0  | 11 | 0  | 0  | 0  | 0  | 0  | 17 | 0  | 23 | 0  |
| 0 | 0 | 7 | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 0  | 22 | 0  |
| 0 | 9 | 8 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 17 | 0  | 21 | 0  |
| 0 | 0 | 0 | 0  | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 19 | 20 | 0  |
| 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

Рисунок 4 – Матрица с суммами пройденных клеток

[7, 8], [7, 1], [4, 1], [9, 1], [16, 1], [16, 2], [11, 2], [11, 3], [11, 4], [11, 5], [11, 6], [11, 7], [11, 8], [11, 9], [11, 10], [11, 11], [12, 11], [13, 11], [13, 12], [13, 13], [13, 14], [16, 13], [9, 13], [6, 13], [7, 13], [7, 14]

Рисунок 5 – Координаты кратчайшего пути

В итоге маршрут пройденного лабиринта выглядит следующим образом (рисунок 6):



Рисунок 6 – Маршрут пройденного лабиринта

## **ЗАКЛЮЧЕНИЕ**

В ходе лабораторной работы был получен опыт написания программ с алгоритмами поиска в ширину и глубину.

## СПИСОК ЛИТЕРАТУРЫ

1. Wikipedia. Поиск в глубину. [Электронный ресурс] – [https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA\\_%D0%B2\\_%D0%B3%D0%BB%D1%83%D0%B1%D0%B8%D0%BD%D1%83](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA_%D0%B2_%D0%B3%D0%BB%D1%83%D0%B1%D0%B8%D0%BD%D1%83) (Дата последнего обращения 08.11.2022).



## ПРИЛОЖЕНИЕ

Ссылка на полный код данной лабораторной работы:

[https://github.com/VeraKasianenko/ITMO\\_ICT/tree/main/1\\_term\\_ICT/  
Algoritms/labs/lab\\_5](https://github.com/VeraKasianenko/ITMO_ICT/tree/main/1_term_ICT/Algoritms/labs/lab_5)