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](#_Toc121403292)

[1 Задание 1 4](#_Toc121403293)

[1.1 Поиск в глубину 4](#_Toc121403294)

[1.2 Поиск в ширину 5](#_Toc121403295)

[Задание 2 6](#_Toc121403296)

[2.1 Лабиринт 6](#_Toc121403297)

[Заключение 7](#_Toc121403298)

[Список литературы 8](#_Toc121403299)

[Приложение 9](#_Toc121403300)

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

# Поиск в глубину

Для поиска в глубину была придумана задача, где необходимо найти сколько существует таких конечных станций метрополитена Нижнего Новгорода, до которых мы может 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).

Изображение выглядит как текст

Автоматически созданное описание

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).



Рисунок 2 – Вывод программы, если человек нашелся



Рисунок 3 – Вывод программы, если никого не нашлось

Задание 2

# 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).

Изображение выглядит как стрела

Автоматически созданное описание

Рисунок 4 – Матрица с суммами пройденных клеток



Рисунок 5 – Координаты кратчайшего пути

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

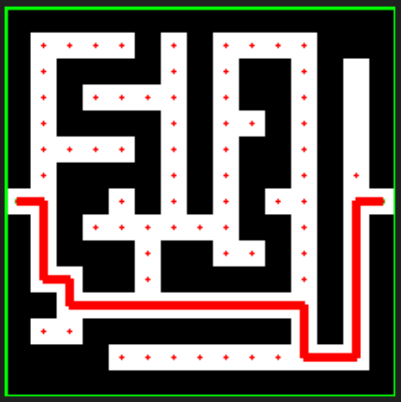


Рисунок 6 – Маршрут пройденного лабиринта

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> (Дата последнего обращения 08.11.2022).

Приложение

Ссылка на полный код данной лабораторной работы:

<https://github.com/VeraKasianenko/itmo_1_sem/tree/main/Algoritms/labs>