

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №4

Выполнили:

Яковлев И.С

Панин Д.В.

Зубов А.А.

Проверил

Мусаев А.А.

23 октября 2023

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Задание — Реализация проверки скобочной последовательности	4
2 Задание — Кратчайший путь метро СПб	5
3 Задание — Реализация алгоритма поиска выхода из лабиринта	6
ЗАКЛЮЧЕНИЕ	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	9

ВВЕДЕНИЕ

Это отчет о выполнении лабораторной работы №4. В первой главе представлено решение первой задачи - реализация алгоритма проверки скобочной последовательности на правильность. Во второй главе представлено решение второй задачи - составлен граф петербургского метро и реализован алгоритм поиска кратчайшего пути от одной станции к другой. В третьей главе представлено решение третьей задачи - реализация алгоритма поиска выхода из лабиринта.

1 Задание — Реализация проверки скобочной последовательности

```
([{}])
True
) (
False 0 символ нарушает последовательность
(([]))
False 3 символ нарушает последовательность

True
```

Рисунок 1.1 — 1 task output

Был написан алгоритм проверяющий скобочную последовательность, содержащую три вида скобок. Алгоритм работает верно, выводит номер первой неправильной скобки (нумерация скобок с 0).

2 Задание — Кратчайший путь метро СПб

```
Введите станцию отправления: Шушары
Введите станцию прибытия: Беговая

При обходе в ширину:
Количество станций: 16
Шушары -> Дунайская -> Проспект Славы -> Международная ->
Time = 0.022976690001087263

При обходе в глубину:
Количество станций: 16
Шушары -> Дунайская -> Проспект Славы -> Международная ->
Time = 0.47422318600001745
```

Рисунок 2.1 — 2 task output

Была перенесена карта станций метро СПб в граф (список смежности). Написан алгоритм поиска минимального пути между станциями с помощью поиска в ширину и поиска в глубину. Также проанализирована скорость работы каждого с помощью модуля `timeit`. Как мы можем видеть алгоритм поиска в ширину работает сильно быстрее при решении этой задачи.

Вывод о выборе нужного алгоритма:

Если мы решаем задачу поиска кратчайшего пути в невзвешенном графе, то эффективней будет использовать поиск в ширину, однако алгоритм поиска в глубину будет эффективней в задачах на поиск всех путей до точки, генерацию всех подмножеств, использование топологической сортировки и поиск в глубину в графах с циклами.

3 Задание — Реализация алгоритма поиска выхода из лабиринта

```
Выход из лабиринта находится на 8 строке в 15 столбце
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 1 0 1 X X X X 1 0 1
1 0 1 1 1 1 0 1 X 1 1 X 1 0 1
1 0 1 0 0 0 0 1 X 1 1 X 1 0 1
1 0 1 1 1 1 0 1 X 0 1 X 1 0 1
1 0 0 0 0 1 0 1 X 1 1 X 1 0 1
1 0 1 1 1 1 0 1 X 1 1 X 1 0 1
X X 1 1 0 1 0 1 X 1 0 X 1 X X
1 X 1 0 0 X X X X 1 1 X 1 X 1
1 X 1 1 1 X 1 1 0 0 1 X 1 X 1
1 X X 1 1 X 1 1 1 1 1 X 1 X 1
1 1 X X X X 1 0 0 0 0 X 1 X 1
1 0 0 1 1 1 1 1 1 1 1 X 1 X 1
1 1 1 1 0 0 0 0 0 0 0 X X X 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Рисунок 3.1 — 3 task output

Алгоритм пройденный путь помечает крестиками, как мы видим путь найден верно, и выведены координаты выхода. Если бы выхода не было, то алгоритм бы вывел, что выхода нет.

```
Выход из лабиринта находится в координатах (4, 4, 5)
[1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1]
[1, 1, 1, 1, 1] ['X', 'X', 'X', 'X', 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1]
[1, 1, 1, 1, 1] [1, 1, 1, 'X', 1] [1, 1, 1, 'X', 1] [1, 1, 1, 'X', 1] [1, 1, 1, 1, 1]
[1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 'X', 'X'] [1, 1, 1, 1, 1]
[1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1] [1, 1, 1, 1, 1]
```

Рисунок 3.2 — 3 task output xyz

Также реализован алгоритм для трёхмерно пространства, ссылка на страничку с кодом в заключении.

Пример был взят для лабиринта размерности 5, так как писать пример для трёхмерного пространства 15x15x15 довольно долго.

ЗАКЛЮЧЕНИЕ

В ходе решения этой лабораторной работы было:

- Написан алгоритм проверки скобочной последовательности на правильность;
- Перенесена карта метро СПб и реализован алгоритм поиска кратчайшего пути между станциями;
- Оценена скорость алгоритмов поиска в ширину и поиска в глубину, на данной задаче;
- Реализован алгоритм поиска пути в лабиринте.

Github с кодами программ — <https://github.com/alekos-bigo/aisd-labs/tree/main/lab-4>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Alfred V. АНО, Jhon E. Hopcroft, Jeffrey D. Ullman Структуры данных и алгоритмы. — Alfred V. АНО, Jhon E. Hopcroft, Jeffrey D. Ullman. Москва, Санкт-Петербург, Киев: Издательский дом "Вильямс". 2003.