Санкт-Петербургский Государственный Технический Университет (Технологический институт)

Кафедра системного анализа и информационных технологий

**Лабораторная работа №6**

Выполнили:

Силуянов Д. К., Степанов Д. А.

Проверил:

Мусаев А. А.

Санкт Петербург

2022

**Задачи:**

1. Написать программу, которая определяет, является ли введенная скобочная структура правильной. Примеры правильных скобочных выражений: (), (())(), ()(), ((())), неправильных — )(, ())((), (, )))), (((). Найдите порядковый номер первого символа (скобки), нарушающего правильность расстановки скобок.
2. Придумайте и решите задачу для алгоритма поиска в глубину. Придумайте и решите задачу для алгоритма поиска в ширину. Объясните, почему для решения поставленных задач были выбраны именно эти алгоритмы поиска (подразумевается возможность выбора и других алгоритмов для решения поставленной задачи).
3. Дана случайная квадратная матрица, заполненная нулями и единицами. Предположив, что 0 – это проход, а 1 – это стена, напишите алгоритм, который найдет выход из лабиринта. Пример:Изображение выглядит как текст, украшен, легкий

   Автоматически созданное описание

**Ход работы:**

1.Мы написали программу, которая определяет, является ли введенная скобочная структура правильной. Под правильной скобочной структурой подразумевается такая структура, которая имеет смысл при построении в предложении. К примеру: ()(()()), ((())), ()((()()())(())) и т.д.

Код программы (рис. 1):

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

рис. 1

Также программа находит порядковый номер первого символа (скобки), нарушающего правильную последовательность расстановки.

2.Мы написали программу, которая использует алгоритм поиска в ширину, используя произвольный граф (рис. 2):Изображение выглядит как устройство, лампа

Автоматически созданное описание

рис. 2

Перенесём этот граф в python в виде словаря (рис. 3): Изображение выглядит как текст, клавиатура

Автоматически созданное описание

рис. 3

Сама функция поиска в ширину выглядит так (рис. 4): Изображение выглядит как текст

Автоматически созданное описание

рис. 4

Зададим начальную и конечную точку (рис. 5):



рис. 5

Завершение кода (рис. 6): Изображение выглядит как текст

Автоматически созданное описание

рис. 6

Итого, консоль выводит следующее (рис. 7): 

рис. 7

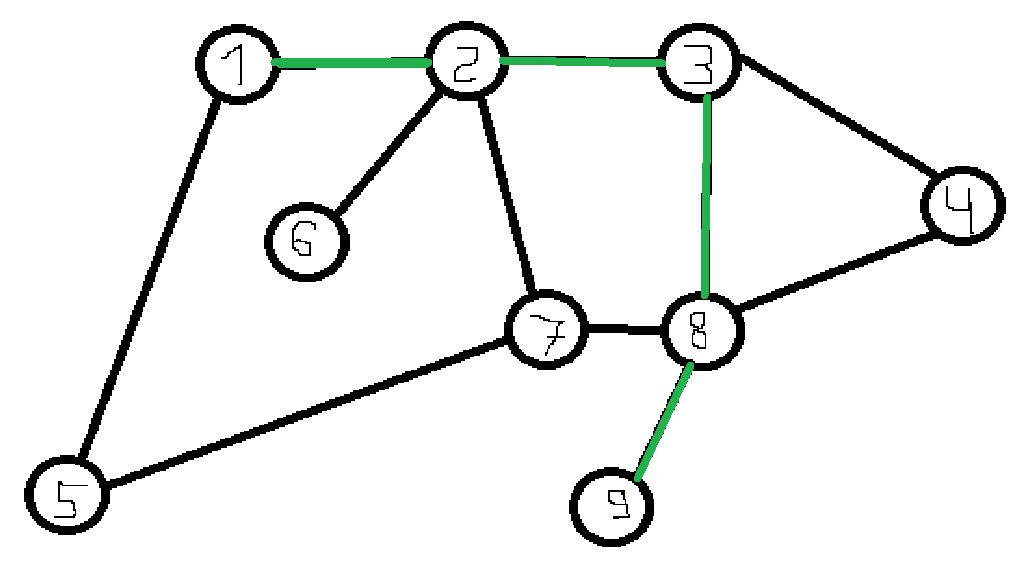
Наглядно, путь от точки 1 до точки 9 выглядит так (рис. 8): 

рис. 8

Мы написали программу, которая использует алгоритм поиска в глубину (рис. 9): Изображение выглядит как текст

Автоматически созданное описание

рис. 9

Итого имеем (рис. 10):



рис. 10

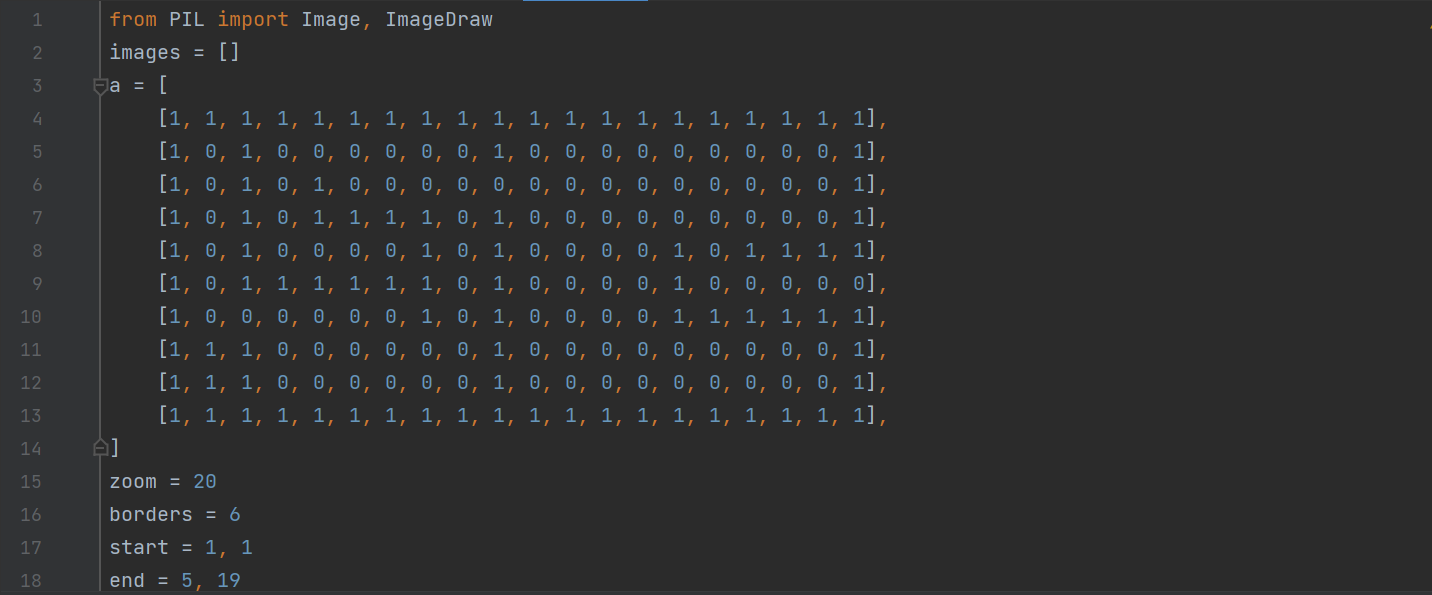
Итак, рассмотрим два метода поиска, рассмотренные выше. Поиск в ширину выгодно использовать, когда важно оптимизировать маршрут, тем самым сделал его наиболее коротким, однако анализ такого маршрута займет достаточно много времени, особенно, если речь идет о большом количестве точек. Поиск в глубину, в свою очередь, зачастую используется в том случае, если нам не важно, каким путем мы попадем в пункт назначения, при этом поиск такого любого маршрута займет куда меньше времени, чем анализ всех возможных путей в поиске в ширину.

Если мы опаздываем на работу, нам важно проанализировать все возможные пути (трамвай, автобус, метро, такси и т.д.) и выбрать наименее времязатратный, чтобы не опоздать очень сильно. Либо можно представить, что мы собираемся в поход. Конечной целью похода станет некоторая поляна. Однако путь лежит через лес, холмы, озера и тропинки. В наших интересах потратить как можно меньше сил и времени на дорогу к поляне. Для этого мы продумаем все пути, возможно воспользуемся картой местности (пример: очевидно, что проще пройти по лесной тропинке, нежели через холмы и пригорки, так мы не потратим силы при подъеме на пригорок). Как раз для таких случаев и используется метод поиска в ширину.

Если мы просто гуляем по городу, у нас нет как таковой цели, также бессмысленно анализировать какие-то маршруты. Мы просто проводим время, гуляя то тут, то там. Гуляем мы при этом по первой попавшейся дороге, проспекту, переулку и т.д. В таких случаях используется метод поиска в глубину. Можно подметить, что роботы-пылесосы тоже используют такой метод, ведь у них нет цели приехать в определённую точку нашей квартиры, ему важно проехать в каждой точке хотя бы один раз, чтобы там убраться, следовательно робот не будет строить маршрут, он, вернее всего, будет ехать прямой, пока не встретит стену.

Итак, существуют ситуации, в которых выгодно использовать тот или иной метод поиска.

3.Мы написали программу, которая находит выход из лабиринта, состоящего из нулей и единиц, где нуль – проход, а единица – стена (рис. 11):

 Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание 

рис. 11

Данная программа использует метод поиска в ширину.