Problem set 2 report

17340027 姚洁倩

一、题目描述

本题中，需要解决的是图的一个最优化问题，题目给出了一张 MIT 的地图，我们需要找到在限定的条件下（如最大户外距离，限定的最大总距离）找到源到目的地的最短路径。使用 DFS 进行实现。

二、解题思路

解决这个问题分成三个步骤：

1. **创建数据结构的表示方法**

   此处，我们需要完成 graph.py 中类的实现。其中，Node 类代表的是节点，Edge 类代表的是边，WeightedEdge 继承的是 Edge 类，代表的是有权重的边，Digraph 类代表的是图。我们需要实现的是 WeightedEdge 和 Digraph 类。

   WeightedEdge

   - 类的成员变量处理 Edge 中的两个之外还有 total_distance 和 outdoor_distance 这两个，加入即可。
   - 两个 get 函数返回相对应的类成员变量的值
   - __str__函数按照一定格式返回有关 weightedEdge 的信息。

   Digraph

   - add_node:首先判断此节点是否已经在图中，若在则 raise 一个 ValueError，若不在则将这个 node 加入进 digraph 的存储 node 的集合中
   - add_edge：首先判断这条边的两个首尾节点是否在图中，若不在，则 raise 一个 ValueError。若都在，则将这条边加入进源节点所映射的 list 中。

2. **建立校园的地图**

   a. **设计自己的图**

   在图中，节点表示的是大楼，边表示的是某两座大楼之间的路径，距离被存储在 WeightedEdge 的 total_distance 和 outdoor_distance 中。

   b. **实现 load_map**

   此函数读取文件，文件每一行的数据都是“源、目的、总距离、户外距离”这样的格式，于是使用一行行读取文件的方法。创建一个 digraph 的实例，每读一行文件就创建两个 node 的实例，分别代表源和目的，将这两个节点加入进图中，使用 try 和 add_node 函数，若出现 ValueError 说明已经加入过了就不再加入。接着创建一个边的实例，同理也尝试将其加入进图中，使用 try 和 add_edge 函数，出现 ValueError 就不加。

   c. **测试 load_map**

   创建了一个 test_load_map.txt 文件，读取这个文件，看所输出的图是否符合预期，测试代码如下：

   ```
   test_graph = load_map("test_load_map.txt")
   print(test_graph.__str__())
   ```

   测试的结果如下：

```
PS C:\coding\Python\17340027-姚洁倩-p2> python .\ps2.py
Loading map from file...
1->2 (10, 9)
2->5 (8, 7)
3->1 (2, 1)
3->2 (3, 2)
4->2 (5, 4)
4->3 (1, 0)
5->4 (6, 5)
```

输出符合预期

3. **使用 DFS 找到最短路径**

a. **目标函数**

使用递归进行深度优先搜索的是 get_best_path 函数，directed_dfs 调用 get_best_path 函数获取最短路径

b. 实现 get_besh_path

使用的是广度优先算法，递归地调用自身。首先判断节点是否合法，不合法则 raise ValueError。接下来判断到现在的节点的户外距离是否超过了可接受的最大户外距离，以及判断到现在的总距离是否比 best_dist 大，若超过了，则没必要继续往下深搜，继续找也找不到最短路径，返回 None 即可。接下来判断 start 节点是否等于 end 节点，若相等，则说明深度优先搜索到了目的节点，此时的 path 参数中的路径和距离都是最优的，返回对应的既可。若不等，则对当前 start 节点的每一个子节点，调用函数 get_best_path，将 start 节点改成子节点，path 改成加入了此子节点的路径信息，其他的传入参数不变，得到返回的新的路径信息，判断返回的信息中的最短路径是否比 best_dist 小，若小，则更新 best_path 和 best_dist。最后返回 best_path 和 best_dist

c. 实现 directed_dfs

调用 get_best_path 函数。使用 try，看看能否返回路径信息，若能返回，则判断返回的路径是否合法，若返回的是 None，说明不存在路径，raise ValueError。若返回的最短路径超过了设定的距离，也要 raise ValueError。若全部都通过，则将最短路径返回。

三、运行结果

Graph.py

```
PS C:\coding\Python\17340027-姚洁倩-p2> python .\graph.py
......
----------------------------------------------------------------------
Ran 6 tests in 0.002s

OK
```

Ps2.py

```
PS C:\coding\Python\17340027-姚洁倩-p2> python .\ps2.py
Loading map from file...
.\ps2.py:111: ResourceWarning: unclosed file <_io.TextIOWrapper name='mit_map.txt' mode='r' encoding=
'cp936'>
  self.graph = load_map("mit_map.txt")
ResourceWarning: Enable tracemalloc to get the object allocation traceback
----------------------
Shortest path from Building 8 to 50 without walking more than 0m outdoors
.Loading map from file...
----------------------
Shortest path from Building 10 to 32 without walking more than 100m total
.Loading map from file...
.Loading map from file...
----------------------
Shortest path from Building 2 to 9
Expected:  ['2', '3', '7', '9']
DFS:  ['2', '3', '7', '9']
.Loading map from file...
----------------------
Shortest path from Building 1 to 32
Expected:  ['1', '4', '12', '32']
DFS:  ['1', '4', '12', '32']
.Loading map from file...
----------------------
Shortest path from Building 2 to 9 without walking more than 0m outdoors
Expected:  ['2', '4', '10', '13', '9']
DFS:  ['2', '4', '10', '13', '9']
.Loading map from file...
----------------------
Shortest path from Building 1 to 32 without walking more than 0m outdoors
Expected:  ['1', '3', '10', '4', '12', '24', '34', '36', '32']
DFS:  ['1', '3', '10', '4', '12', '24', '34', '36', '32']
.Loading map from file...
----------------------
Shortest path from Building 32 to 56 without walking more than 0m outdoors
Expected:  ['32', '36', '26', '16', '56']
DFS:  ['32', '36', '26', '16', '56']
.Loading map from file...
```

```
----------------------
Shortest path from Building 32 to 56
Expected:  ['32', '56']
DFS:  ['32', '56']
.
----------------------------------------------------------------------
Ran 9 tests in 0.074s

OK
```

运行的测例全部通过