CS2020: Data Structures and Algorithms (Accelerated)

Problems 18–18

Released: Tuesday, March 22th 2011, Due: Wednesday, March 30th 2011, 13:59

Overview. You only have one programming task this week (due to the fact that you need to study for the upcoming Quiz 2). The task for this week is very (?) similar with last week :O, is it?

Collaboration Policy. As always, you are encouraged to work with other students on solving these problems. However, you **must** write up your solution **by yourself**. In addition, when you write up your solution, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). Any deviation from this policy will be considered cheating, and will be punished severely, including referral to the NUS Board of Discipline.

Problem 18. (Maze Navigation - Version 2)

Now, what if the maze is a little bit more realistic. On top of characters '#', '!', and 'E' as defined in the previous task, now each pass-able cell have label '1', '2', '3', ..., '9' that denotes the *cost* of passing those cell. Cell labeled with '!' has cost '0'. Cell(s) labeled with 'E' has cost '1'.

Everything else are the same as the previous task (Problem 17), but now you have to identify the *exit with minimum cost* (not necessarily the nearest) and the number of steps to reach that exit with minimum cost. Again, if there is no way you can reach a cell with character 'E', you have to output -1 (to say 'impossible') as your answer.

Several examples:

`maze5 - the minimum cost: 17, only one exit at row 1, column 0`

```
##########
E11111111#
########1#
#!1111111#
##########
```

`maze6 - there are two paths from '!' to 'E' at row 0, column 4, the minimum cost: 16, by taking left route: 7x2 + 1 + 1 = 16`

```
####E####
#2221333#
#2#####3#
#222!333#
#########
```

`maze7 - there are two reachable exits, the minimum cost: 10, by taking right route: 3x3 + 1 = 10 to exit 'E' at row 3, column 8`

```
####E####
#2221333#
#2#####3#
#222!333E
#########
```

`maze8 - there are two reachable exits, the minimum cost: 9, by taking left route: 8x1 + 1 = 9 to exit 'E' at row 0, column 4. Note that the nearest exit 'E' at row 3, column 8 has cost: 3x9 + 1 = 28, which is not the minimum.`

```
####E####
#1111999#
#1#####9#
#111!999E
#########
```

```
maze9 - there is no reachable exit, output -1.

#########
#!#11111E
#########
```

Again, to get you started, please answer the following questions. You have to write the answers as comments in `Maze2.java`.

- Graph Modeling

  1. What are the vertices and the edges of this implicit graph?
  2. How are you going to represent the 'weight'? Is it on the vertex or is it on the edges?
  3. How many edges and how many vertices are there in an `R × C` maze?
     You do not have to count the precise number! Use big O notation in terms of R and C.
  4. What is the graph problem that we are trying to solve here?

- Graph Algorithm
  Remember that this time: if we can reach more than one possible exits, we must select the one with the minimum cost (not necessarily the nearest one!)
  If there are several possible algorithms, choose the one with the fastest asymptotic running time. Marks will be deducted for implementing the inferior/wrong algorithm.
  There is no need to give formal proof why a certain algorithm can or cannot be used.
  Few logical sentences should be enough.

  1. Can you use $O(V + E)$ Depth First Search (DFS) to solve this problem? Why?
  2. Can you use $O(V + E)$ Breadth First Search (BFS) to solve this problem? Why?
  3. Can you use $O(E \log V)$ Kruskal's algorithm to solve this problem? Why?
  4. Can you use $O(VE)$ Bellman Ford's algorithm to solve this problem? Why?
  5. Can you use $O((V + E) \log V)$ Dijkstra's algorithm to solve this problem? Why?

- Implementation: `Maze2.java`
  Remember: never ever touch the `main` method to avoid unnecessary input/output errors.
  The skeleton code written by Steven can already read in the maze(s) and output the result in the correct format. You have to implement the following method: `iii MinCostExit()`. The requirements for this method is explained in `Maze2.java`. Class 'iii' is used to model Integer triple. Hopefully you do not encounter too many technical issues as the format is exactly similar to last week's PS7.