SNU CSE: Data Structure Due Date : 11 pm, Dec 12 2018

Assignment 5

Lecturer: Prof. Satti

1. An edge-weighted undirected graph G is given as a grid graph of size 20×20, with 40 vertices removed (hence G has n=360 vertices). Each vertex of the graph has an integer coordinate (*x*, *y*) where (0 ≤ *x* < 20, 0 ≤ *y* < 20). Each vertex also has a height value h (0 ≤ h<7), where h value is given in the `height.txt` file that is uploaded at the ETL. The i-th line of the height.txt file contains the height information of (i,0), (i,1), (i,2)… and (i,19). If the height of any vertex is given as -1, it means that the vertex is one of the 40 removed vertices. For any two non-removed vertices (x1, y1) and (x2, y2), there is an edge between them if and only if |x1 −x2| +|y1 − y2| ≤ 2. Then the distance d between (x1, y1) and (x2, y2) is given by the equation: $d^2 = (h1 - h2)^2 + (|x1 - x2| + |y1 - y2|)^2$ .The weight of this edge is defined as the distance between the two vertices. Write a program `init.java` that reads the data `height.txt` and implements the graph G as a linked list.

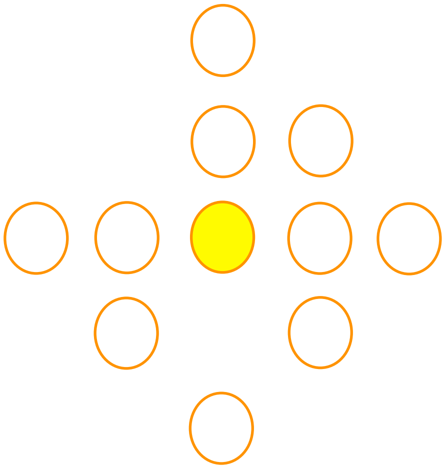
가로세로 거리가
2 이하면 엣지 존재.
weight은 x-y 길이
와 높이가 이루는
삼각형의 빗변

2. Let vertex (0,0) be the source vertex. Write a program `Dijestra.java` that implements Dijkstra's algorithm using the linked list representation created in Problem 1, and using a min-heap implementation.
You are allowed to use additional array structures (for example, an array to keep the predecessor vertex information for each vertex). Output the computed shortest path information from the source vertex to all vertices. Your output results for each destination vertex v should include distances and edge lists of the shortest path from the source vertex to the v. Do not include removed vertices in your output. The format of your output should be as shown in the following example:

min-heap은 priority queue라
SP(혹은 d)를 불러오는 데 쓸 수 있을 듯

Ex) Output_b.txt

```
(0,0)-(1,2): 4.12 (0,0) (1,1) (1,2)
(0,0)-(2,1): 2.5 (0,0) (1,1) (2,1)
(0,0)-(2,2): 6.15 (0,0) (1,1) (1,2) (2,2)
...
```



3. Make a program `prime.java` that implements the Prim's algorithm to compute the minimum spanning tree using the linked list representation created in Problem 1 and a min-heap. Use vertex (19,19) as the initial seed vertex.

You are allowed to use additional array structures. Output the edge list of the tree and the total weight of the minimum spanning tree. Output one edge information per line. The format of your output should be as shown in the following example:

Ex) Output_c.txt

```
(1,2) (2,2)
(8,12) (8,13)
…
(19,19) (20,20)
Weight: 1244.14
```

Assessment

Your codes will be judged on its correctness. Please make sure that your code runs on Linux terminal. TA will not accept any complain if your code doesn't run on Linux terminal correctly, even if it runs on other IDEs or working environment correctly.

Submission

Compress all your source and output files into a single file as your-full-name_SID.zip.