CS2040S: Data Structures and Algorithms

Tutorial Problems for Week 13 Monday: Shortest Paths (II)

For: 7 November 2022, Tutorial 11

Problem 1. True or False?

For each of the following, determine if the statement is True or False, justifying your answer with appropriate explanation.

a) If a undirected graph has a unique minimum spanning tree, and we run Prim's and any single source shortest path algorithm from the same source, the final result of edges chosen will form the same spanning tree.

b) If the weights of all edges in a positively weighted graph are unique, there is always a unique shortest path (the smallest total cost is unique) from a source to a destination in such a graph.

c) A connected, undirected graph will always form a shortest path tree with $V - 1$ edges.

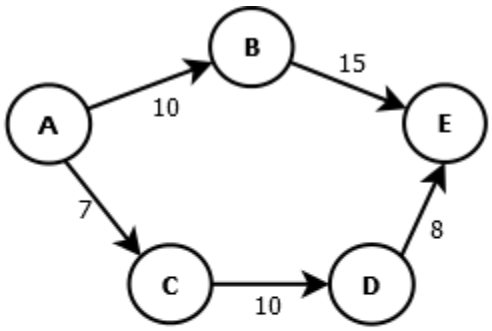
d) The shortest path from any vertex A to any vertex B is the combination of the shortest path from A to any vertex C and shortest path from C to vertex B.

e) The algorithm below will still give the correct shortest path for positively weighted graphs.

```
function Dijkstra(Graph, source):

    create vertex set Q

    for each vertex v in Graph:
        dist[v] = INFINITY
        prev[v] = UNDEFINED
        add v to Q
    dist[source] = 0

    while Q is not empty:
        u = vertex in Q with min dist[u]

        remove u from Q

        for each neighbor v of u: // only v that are
           still in Q
            alt = dist[u] + length(u, v)
            if alt < dist[v]:
                dist[v] = alt
                prev[v] = u
    return dist[], prev[]
```

Problem 2. Promoter

A salesman frequently needs to drive from one city to another to promote his products. Since time is of the essence, he wants to use the shortest route to get from one city to another. However in every city he passes he will have to pay a toll fee. The toll fee is the same for every city and it is a positive number. Therefore, given two different routes of the same distance (positive numbers) to get from city A to city B, he will prefer the one which passes through fewer cities. An example is shown below. Propose the best algorithm using what you have learnt so far (and a bit more), so that the salesman will choose a route from any source city A to any destination city B such that it has the shortest distance and also passes through the fewest cities (you may assume there is a way to get from any city A to any city B). What is the running time for your algorithm?



To get from A to E, route A,B,E is preferred over route A,C,D,E even though both have the same cost 25, since A,B,E goes through fewer cities.

Problem 3. Lego Mindstorms EV3 (2013/2014 CS2010 S1 WQ2)

Steven has a new toy, LEGO Mindstorms EV3 (which he will pass to his daughter Jane in a few years' time). He wants to command his robot to move from one cell to another cell in an $m \times n$ grid containing mostly '`.`' (passable cells) and some '`#`' (blocked cells). The diagram below shows a sample 4×7 grid with 8 obstacle cells (the '`#`'s):

```
. . . . . . .
. # # # # # .
. # . . . # .
. . . # . . .
```

Initially, Steven's robot is at coordinate (0, 0), the top-left corner of the grid, and faces east. At each cell, Steven's robot can only do one of the two actions below:

1. Move forward by one cell according to it's current direction. For example, if the robot currently at coordinate (0, 0) and faces east, it will be in coordinate (0, 1) and still faces east after this action. Such action consumes 3 seconds.
2. Rotate the robot by 90 degrees clockwise (turn right); the robot stays in the current cell. For example, if the robot currently at coordinate (0, 0) and faces east, it will still be in coordinate (0, 0) but now faces south after this action. Such action consumes 2 seconds.

Give an algorithm that Steven can use to upload to his robot so that his robot can move from coordinate (0, 0) to coordinate $(m-1, n-1)$ using the minimum amount of time (you can assume there is at least 1 such path)! Of course, Steven's robot cannot go outside the $m \times n$ grid and cannot move to a blocked cell throughout the execution of the program. When the robot stops at coordinate $(m-1, n-1)$, it can face any direction.

On the sample grid above, the best robot path is as indicated below with a total time of 29 seconds, i.e. move forward 6 times (18 seconds), turn right (2 seconds), and then finally move forward 3 times (9 seconds). On another sample grid below, the best robot path is as indicated below with a total time of 20 seconds, i.e. turn right (2 seconds), move forward 2 times (6 seconds), turn right 3 times so that the robot faces south $\rightarrow$ west $\rightarrow$ north $\rightarrow$ east (6 seconds), and finally move forward 2 times (6 seconds).

```
. # #
. # #
. . .
```

Convert the following into a graph problem by answering the following questions.

Problem 3.a. What do the vertices and the edges of your graph represent?

Problem 3.b. What is the upper bound of the number of vertices and edges in your graph?

Problem 3.c. What is the graph problem that you want to solve?

Problem 3.d. What is the most appropriate graph algorithm to solve this problem and it's time complexity? (Note: What should the terms in the time complexity be expressed as?)

Problem 4. Friends meeting up (2015/2016 CS2010 S2 WQ2)

Two friends lives in cities A and B respectively, and they want to meet up. In order to convenience the both of them, they come up with a strategy to meet up at a city C, where C satisfies the following 2 conditions:

1. The sum of their total travelled distance to C is the smallest.
2. The absolute difference between the distance travelled by each person is minimized.

Note that the **first condition is more important than the second**. Come up with an efficient algorithm to find the city they should meet up in. If there are multiple candidate cities which satisfy the above criteria just output one of them. You can assume that there exist at least 1 such city C and also the roads and cities can be represented as an undirected connected weighted graph, where the edge weight is the distance between the connected cities.

Problem 5. Multiple-Choice Questions

Problem 5.a. Which of the following properties of a Binary Search Tree (BST) is false?

a) The right descendants of the root will contain keys larger than the key of the root for BSTs with unique keys.

b) Every node of a BST will have two child nodes..

c) The time complexity of inserting into a BST is $O(n)$ time.

d) The left child of the root has a key that is smaller than the key of the root.

Problem 5.b. Consider a connected, undirected graph G which can have two or more cycles, and a vertex X in the graph. Which of the following statements is true?

a) G will always have more than one possible minimum spanning tree.

b) The shortest paths from X to all other vertices are *always* unique.

c) We can obtain a spanning tree of G in $O(V + E)$ time.

d) The Bellman-Ford algorithm applied on G from X will never produce the correct shortest paths.

Problem 5.c. Mark has designed a graph-based game. He will first create a directed unweighted graph with V vertices and E edges. He will then pick a source vertex s, a destination vertex d, and some number K. He will now start from s and place K on s. If s has M outgoing edges, he will divide K by M (integer division), and so each neighbor of s will have the value K/M placed on them. This will continue for each vertex with a value placed on them and will stop if d is reached, or the number K' placed on the vertex v is such that $K'/$(number of outgoing edges of v) $== 0$. Mark wants to know what is the minimum value of K so that based on his game he can reach d. You can assume that there is always a path from x to y.

The best algorithm to find the minimum value of K is equivalent to

a) solving a MST problem with time complexity $O(E \log V)$

b) solving a SSSP problem with time complexity $O((V + E) \log V)$

c) solving a SSSP problem with time complexity $O(VE)$

d) solving a SSSP problem with time complexity $O(V + E)$

e) modelling the graph and computing all possible paths from s to d and find the minimum cost path with time complexity $O(V!)$