CS 575 Theory Assignment 3.3 – Spring 2020

- Due: 4/14/2020 (on 11:59pm)
- Total possible points: 65

1 [25 points]

Jack plans to drive from city A to city B along a highway. Suppose $g_0, g_1, g_2, \ldots, g_k$are the gas stations along the highway and are ordered in increasing distance (in miles) from city A, where g_0 is located at the starting place (city A). Let d_i be the distance (in miles) between g_i and $g_i + 1, i = 0, \ldots, k{-}1$, and we assume that these distances are known to Jack. Each time Jack fills gas to his car, he gets a full tank of gas. Jack also knows the number of miles, m, his car can drive with a full tank of gas. Jack's goal is to minimize the number of times he needs to stop at gas stations for his trip.

a. [8 points]

Design a greedy algorithm to solve the above problem, i.e., to minimize the number of stops for gas.

Answer

A greedy algorithm is to drive as far as possible without running out of gas before the next gas station. More specifically, when Jack starts from a gas station with a full tank, he computes the farthest gas station g_i he can go (i.e., if he attempts to drive to the next gas station g_{i+1}, the car will run out of gas before reaching g_{i+1}).

b. [12 points]

Show that your greedy algorithm has the greedy choice property, i.e., each local decision will lead to the optimal solution. Basically you need to argue for the correctness of your algorithm.

Answer

Locally optimal choice lead to globally optimal solution.

This problem also has the greedy-choice property. Suppose there are r gas stations beyond the start that are within m miles of the start. The greedy solution chooses the rth gas station as its first stop. No station beyond the rth works as a first stop, since Jack would run out of gas first. If a solution chooses a gas station j < k as its first stop, then Jack could choose the rth gas station instead, having at least as much gas when he leaves the rth gas station as if he'd chosen the jth gas station. Therefore, he would get at least as far without filling up again if he had chosen the rth gas station. The same argument can be made after each stop.

c. [5 points]

What is the running time of your algorithm?

Answer

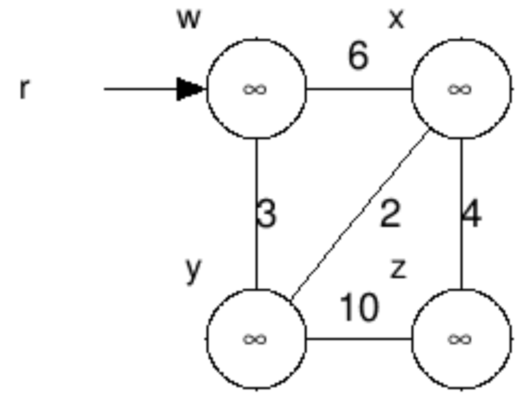
If there are n gas stations, Jack needs to consider each one just once to figure out the gas stations he needs to stop for gas. The running time is O(n).

2. [20 points]

Given the Prim's algorithm shown below (a min-priority queue is used in the implementation):

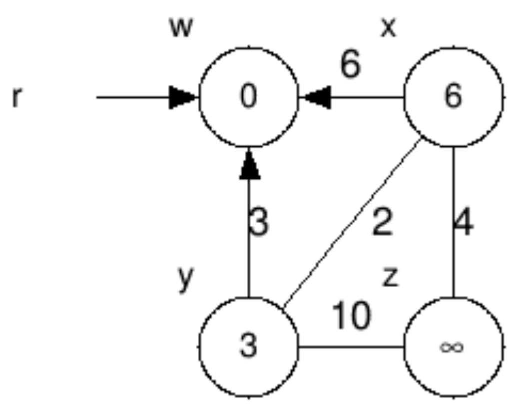
Apply the algorithm to the weighted, connected graph below (the initialization part has been done). Show a new intermediate graph after each vertex is processed in the while loop. For each intermediate graph and the final graph, you need to show the vertex being processed, the new key value for each vertex and edges in the current (partial) MST (draw a directed edge from vertex v to u if v.π = u).

```
PRIM(G, w, r)
  Q = ∅
  for each u ∈ G.V
    u.key = ∞
    u.π = NIL
    INSERT(Q, u)
  DECREASE-KEY(Q,r,0)    // r.key =
0
  while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
      if v ∈ Q and w(u,v) < v.key
        v.π = u
        DECREASE-KEY(Q, v, w(u,v))
```

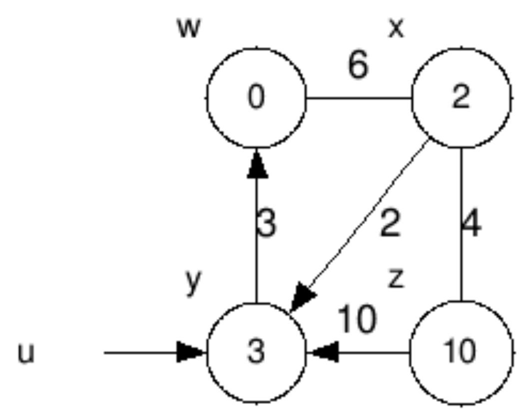


Answer

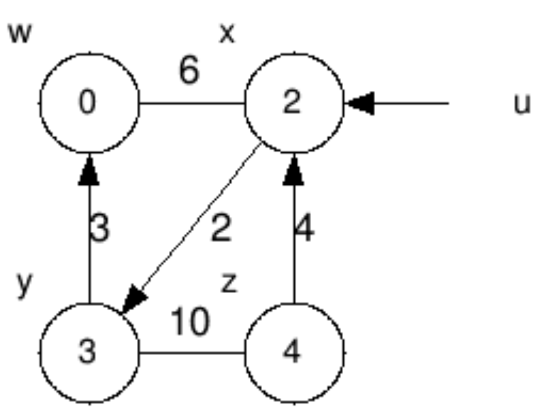
After w is processed:



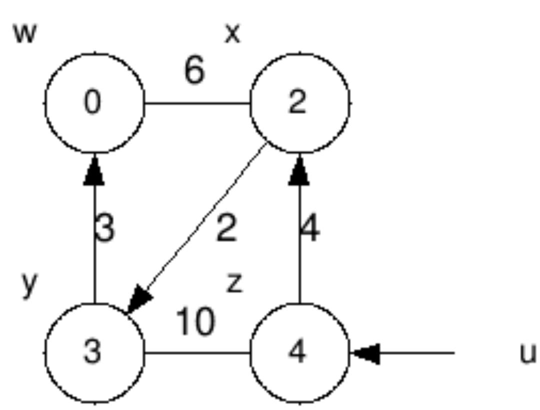
After y is processed:



After x is processed:



After z is processed (the final graph):



3 [20 points]

Apply Kruskal's algorithm to the graph below. Show new intermediate graphs with the shaded edges belong to the forest being grown. The algorithm considers each edge in sorted order by weight. An arrow points to the edge under consideration at each step of the algorithm. If the edges joins two distinct trees in the forest, it is added to the forest, thereby merging the two trees.