Saad Saiyed | 200415258

Assignment 8

Problem 1 (6+1 marks).

(a) textbook, Problem 9.15(a) For each algorithm, give a list of the edges in the order in which they are added to the MST.

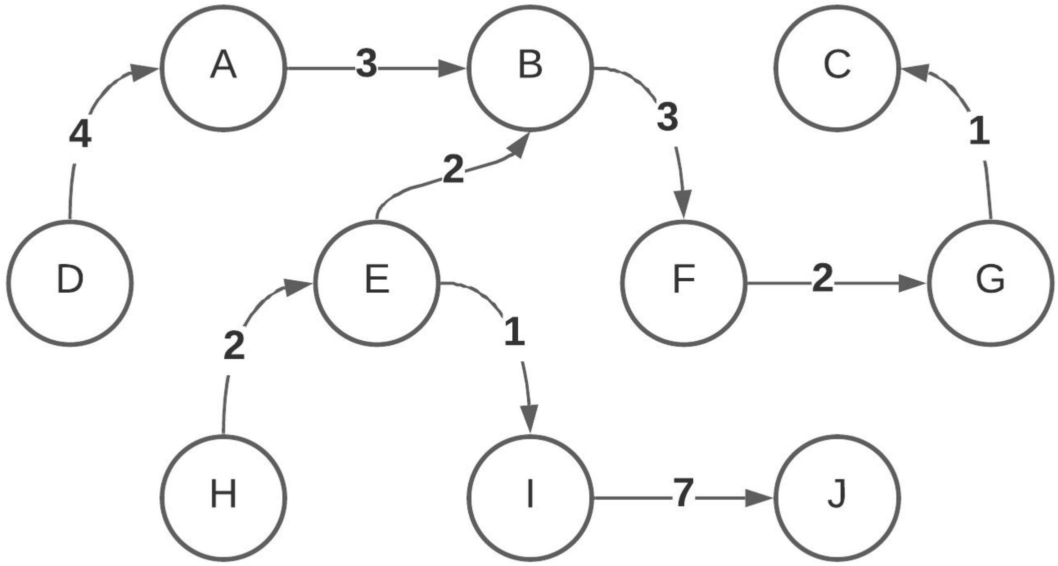
(b) textbook, Problem 9.15(b)

Ans:

(A) Problem 9.15(a)

a. The steps to construct a new MST using Prim's algorithm is as follows;
    i. Choose one node from the old tree.
    ii. While the new tree has less than n (number of nodes in old tree) nodes.
        1. Look for the smallest edge connecting to the nodes of existing new trees.
        2. Connect that smallest value edge node to the new tree.
    iii. End the loop

At the end of the loop we will get new tree like following with edges which are in order -> ***A, B, E, I ,H, F, G, C and J***



b. The steps to construct a new MST using Kruskal's algorithm is as follows;
   i. Create a forest of all the edges with connecting nodes.
   ii. Sort this forest with minimum edge value.
   iii. Until all the edges from this forest is not over
      1. Add the two nodes in the new tree taking the values from the minimum value of edges from the forest.
      2. Reject the edges which create loops.
      3. Remove edge value from the forest.
   iv. End the loop.

At the end we will get the same result as from the Prim's algorithm (For this example). Although the forest array will look something like this; (Unsorted and with the occurrence of each edge on top-left)

6 A -> B = 3	8 A -> D = 4	A -> E = 4
B-> C = 10	7 B -> F = 3	3 B -> E = 2
1 C -> G = 1	C -> F = 6	D -> E = 5
D -> H = 6	4 E -> H = 2	2 E -> I = 1
E -> F = 11	5 F -> G = 2	F -> J = 11
F -> I = 3	G -> J = 8	H -> I = 4
9 I -> J = 7		

Problem 2 (2+2 marks). Both (a) and (b) concern the graph in Figure 9.82 (page 438) in your textbook. In both cases, if your traversal algorithm at any point in time has more than one option of which vertex to visit next, choose the alphabetically smaller vertex. (a) Show the list of the vertices of the graph, as they would be traversed by BFS starting at vertex A. (b) Show the list of the vertices of the graph, as they would be traversed by DFS starting at vertex A.

Ans: Considering the fact that at any point in time has more than one option of which vertex to visit next, choose the alphabetically smaller vertex, we perform;

a. For BFS, we start at A and look at the nodes B, C, D.
   i. We explore till F and at the end we get the order as: **A, B, C, D, E, G and F**

b. For DFS we explore the graph in a depth measuring fashion. We use a stack which holds the information of which nodes need to be visited next and how many times each node has been visited with their end time. For example in this example for node A the visiting count is 1 and end time is 14.
   i. So after we reach G as the last node through this process we get our final order as: **A, B, C, D , F, E and G**

Problem 3 (4 marks). Provide an O(|V |) algorithm that solves the following problem. given: a directed graph G = (V, E), represented as an adjacency matrix. task: determine whether or not there is a vertex w ∈ V such that, for every other vertex v ∈ V with v 6= w, we have: (v, w) ∈ E but (w, v) ∈/ E .

Ans: We have G = (V, E) where v = { v1, v2, v3, ..., vn}

For each w ∈ V,

Problem 4 (3 marks). Show that the decision problem P is reducible to the decision problem P 0 , using a reduction mapping red that runs in time polynomial in the size of its input, i.e., red(N) = O(f(N)) where f is some polynomial in N. Problem P: given: an undirected graph G = (V, E) and a number k ∈ N, where k ≤ |V | question: is there a set V 0 ⊆ V of at least k many vertices such that, for all v, w ∈ V 0 , the (undirected) edge (v, w) belongs to E? Problem P 0 : given: an undirected graph G = (V, E) and a number k ∈ N task: is there a set V 0 ⊆ V of at least k many vertices such that there are no two vertices v, w ∈ V 0 for which the (undirected) edge (v, w) belongs to E? Recall that the size of a graph G = (V, E) is |V | + |E|.

Ans: The decision problem P states:

Given a graph G and a number Ko, does G contain a vertex cover of size at least Ko?

Where the vertex cover is a set V0<=V such that for all v, w ∈ V0 the (v,w) edge belongs to E.

The decision problem P0 states;

Given a graph G and a number Ko, does G contain an independent set of size at least K?

Where an independent set is a set V0<=V such that there are no vertices v,w ∈ V0 for which (v,w) ∈ E.

Lemma:

Let G=(V,E) be a graph. Then S is an independent set if and only if its complement V-S is a vertex cover.

As S is an independent set, consider any arbitrary edge e = (u,w) since S is an independent set, both u and v can't be in the set S. So one of them must be in (V-S). Therefore, it follows that every edge has at least one edge in (V-S) and thus (V-S) is a vertex cover.

Conversely, suppose V-S is a vertex, now considering an edge e=(u,w), neither of which end vertices lie in V-S, contradicts an assumption that V-S is a vertex cover. It follows that no two nodes of S are joined by an edge and hence S is an independent set.

To show that decision problem P is polynomial time reducible to decision problem PO:-

From lemma(stated previously) we know

Maximum independent set + maximum vertex cover = |V|

(Where V is the vertex set of G)

So if we can find a solution for the decision problem P of size at least Ko, then rest of |V|-Ko vertices will be a solution to the decision problem Po, which is indeed polynomial time reduction, as Po is the complement of P.