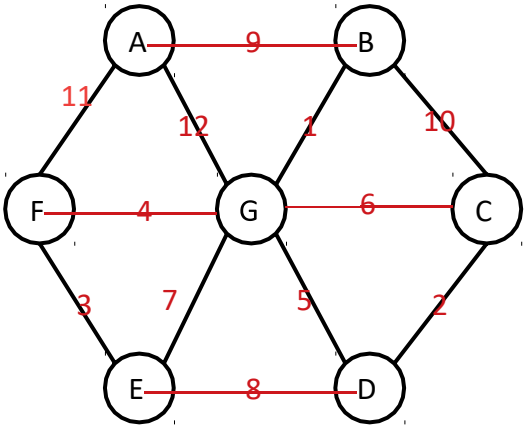
CS 3353 – Spring 2020 – The Online Edition

Homework 01 – Spanning Trees

1. Use the following graph to answer the questions that follow.



a) Following Prim's algorithm as described in lecture, list the edges as they are added by the algorithm when starting with Node G.

(G – B), (G – F), (F – E), (G – D), (D – C), (B – A)

b) Following Prim's algorithm as described in lecture, list the edges as they are added by the algorithm when starting with Node A.

(A – B), (B – G), (G – F), (F – E), (G – D), (D – C)

c) Following Kruskal's algorithm as described in lecture, list the edges as they are added by the algorithm when starting with Node D.

(B – G), (C – D), (F – E), (F – G), (D – G), (A – B)

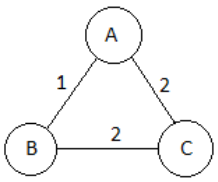
d) Following Kruskal's algorithm as described in lecture, list the edges as they are added by the algorithm when starting with Node F.

(B – G), (C – D), (F – E), (F – G), (D – G), (A – B)

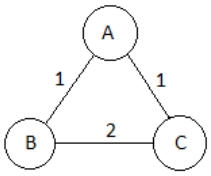
e) Modify Kruskal's algorithm to find the Maximum Spanning tree. Then do parts (c) and (d) again to find the Max Spanning Tree of the graph above.

(A – G), (A – F), (B – C), (A – B), (D – E), (E – G)

2. Construct a graph that contains exactly three vertices and has exactly two minimum spanning trees.



3. Construct a graph that has exactly two edges of the same weight, but has exactly one (unique) minimum spanning tree. Your graph can have at most 4 vertices.



4. Related to Programming Assignment 03 – Kruskal's Algo Implementation, briefly describe your choice for "more efficient" implementation of a disjoint set collection. In particular, succinctly describe how Find and Union will be implemented.

For my theoretical implementation, the collection will be represented by two hash tables. The first hash table will have keys be the elements of the collection, and the values will be an identifier of the subset they belong to. The second hash table will have keys be the subset identifiers, and the keys will be vectors containing elements.

The Find function takes the argument element and uses the first hash table to retrieve its subset identifier.

The Union function takes both argument elements and finds their subset identifiers using the first hash table. These subset identifiers are then plugged into the second hash table to retrieve the vectors representing the subsets. One vector will have all of its elements inserted into the other vector and then cleared. Each element of the cleared vector will have its subset identifier changed in the first hash table.