

**Question 2 [25 points] Design an algorithm to compute the maximum spanning tree of a graph.**

A simple idea to solve this problem to get the following two equivalent equations.

Finding the maximum spanning tree of a graph = Finding the minimum spanning tree of the same graph where all the edge costs are negated.

As a result, the first step is to negate all the edges in the graph. Then we can apply one of the algorithms that we have learned in class. I will use kruskal's algorithm.

Below is the general steps of kruskal's algorithm: (not in detail)

1. Create an array E where each edge is ordered by its weight. ( Suppose we use merge sort)
2. For each edge in the set ( the weight is in increasing order), connect them if the do not form a cycle in the spanning tree formed now.
3. Keeping doing step 2 until the graph is a spanning tree.

Therefore, we first negate all the edges and apply kruskal's algorithm to get the maximum spanning tree of graph.

The running time of this algorithm is :

$$O(E) + O(E \log(E)) = O(E \log(E))$$

Negate the edges      Time Complexity for Kruskal's algorithm

Note that In Kruskal's algorithm, sorting takes  $O(E \log(E))$  and union-find takes  $O(\log(E))$  time.