

8. Show how to find the *maximum* spanning tree of a graph, that is, the spanning tree of largest total weight.

We will solve this problem by using a slightly modified version of Kruskal's algorithm. In short, we will first negate the weights, and then apply Kruskal's as usual.

Algorithm:

```
1  kruskal_max(G, w){
2
3      for all  $u \in V$ :
4           $w[u] = -w[u]$ 
5          makeset(u)
6
7       $X = \{\}$ 
8      sort the edges  $E$  by weight
9      for all edges  $\{u, v\} \in E$ , in increasing order of weight:
10         if  $\text{find}(u) \neq \text{find}(v)$ :
11             add edge  $\{u, v\}$  to  $X$ 
12             union( $u, v$ )
13 }
```

Let us justify why this works. First, we assume that Kruskal's algorithm is understood and works as expected. That is to say, we assume it produces the minimum spanning tree given an input graph and edge weights. We also assume that we are considering a graph containing only positive, non-zero weights.

By negating each of the weights, the order is essentially reversed. A sort in increasing order will now have the precise opposite order as before the negation occurred. This is easily seen by considering absolute values. When considering positive numbers, the largest numbers are those that have the largest absolute values. With negatives, this is the opposite: the larger the absolute value, the smaller the number.

As we can assume that the unmodified Kruskal's algorithm works, then we have that the output of the algorithm is the spanning tree with the smallest possible total weight. As all the weights are negative, each possible total weight is necessarily negative. Thus, the minimum weight will be the weight with the highest absolute value. As we have shown, the weights with maximum absolute value are the largest in the positive case. Thus, the minimum spanning tree with negated edges necessarily has the highest absolute value of all spanning trees, and thus is the maximum spanning tree in the positive case.