Mini Homework 8

1. Consider Kruskal's algorithm, sort edges into non-increasing order, and take one by one in non-increasing order.
2. **Algorithm**(G, k):
    1. $T_{\min}$:= minimum spanning tree of G.
    2. $T_{\max}$:= maximum spanning tree of G.
    3. **if** $T_{\min}$ .weight $\leq k \leq$ $T_{\max}$ .weight: **return** true.
    4. **else**: **return** false.
3. If k is outside the range of [$T_{\min}$ .weight,$T_{\max}$ .weight], then a spanning tree of weight k is not possible, since $T_{\max}$ is maximum, and $T_{\min}$ is minimum. Otherwise, k is within the range, then the spanning tree of weight k can be generated as follows.
   Suppose there is a non-maximum spanning tree $T = (V, E)$ of weight i, where $T_{\min}$ .weight $\leq i <$ $T_{\max}$ .weight. The goal is to substitute an one-edge for a zero-edge so that a spanning tree $T' = (V', E')$ of weight $i + 1$ is generated.
   Consider the set of zero-edges $E_0 \subseteq E$. Removing an edge in E_0 partitions T into two subtrees T_1 and T_2. Let G_1 and G_2 be the subgraphs induced by the vertices of T_1 and T_2, respectively. The vertices of G_1 and G_2 form a cut of G with some edges crossing G_1 and G_2. In so doing, there must be a corresponding cut for every edge in E_0.
   Next, there must exist an edge $e_0 \in E_0$ such that its corresponding cut contains an one-edge e_1. If there is no such cut, then it is not possible to update a spanning tree of weight i to weight $i + 1$. However, this is a contradiction to that a spanning tree of weight i is not maximum. Finally, e_0 can be replaced with e_1 so that the resulting graph is a spanning tree of weight $i + 1$.
   By inductive proof, any spanning tree of weight $k \in$ [$T_{\min}$ .weight,$T_{\max}$ .weight] is possible.