Algorithms and programming
École normale supérieure - Département d'informatique
TD 8: Minimum Spanning Tree

---

1. Let (u, v) be a minimum-weight edge in a graph G. Show that (u, v) belongs to some minimum spanning tree of G.
2. Is the following correct ? Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V - S)$ be any cut of G that respects A, and let (u, v) be a safe edge for A crossing $(S, V - S)$ (that is, $(u, v) \cup \{A\}$ is a subset of some minimum spanning tree for G as well). Then, (u, v) is a light edge for the cut.
3. Show that if an edge (u, v) is contained in some minimum spanning tree, then it is a light edge crossing some cut of the graph.
4. Let e be a maximum-weight edge on some cycle of $G = (V, E)$. Prove that there is a minimum spanning tree of $G' = (V, E - e)$ that is also a minimum spanning tree of G.
5. Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut (light edge = edge of the minimum weight). Show that the converse is not true by giving a counterexample.
6. Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree T' of G, the list L is also the sorted list of edge weights of T'.
7. Let T be a minimum spanning tree of a graph $G = (V, E)$, and let V' be a subset of V. Let T' be the subgraph of T induced by V', and let G' be the subgraph of G induced by V'. Show that if T' is connected, then T' is a minimum spanning tree of G'.
8. Kruskal's algorithm can return different spanning trees for the same input graph G, depending on how ties are broken when the edges are sorted into order. Show that for each minimum spanning tree T of G, there is a way to sort the edges of G in Kruskal's algorithm so that the algorithm returns T.
9. Suppose that the graph $G = (V, E)$ is represented as an adjacency matrix. Give a simple implementation of Prim's algorithm for this case that runs in $O(|V|^2)$ time.
10. Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Kruskal's algorithm run ? What if the edge weights are integers in the range from 1 to W for some constant W ?
11. Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Prim's algorithm run ? What if the edge weights are integers in the range from 1 to W for some constant W ?