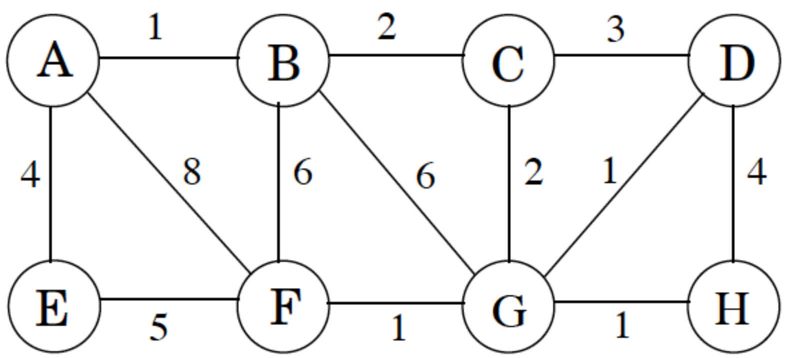
CMPSC 465	Data Structures & Algorithms	
Fall 2025	Ke Chen and Yana Safonova	Worksheet 8 Solution

Monday, Oct 27, 2025

1. **Minimum Spanning Trees.** Run Prim's Algorithm to find a minimum spanning tree for the following graph. Whenever there is a choice of nodes, always use alphabetic ordering (e.g. start from node A). Show the order edges are added and the weight of the partial MST at each step.



Solution:

Vertex included	Edge included	Cost
A		0
B	AB	1
C	BC	3
G	CG	5
D	GD	6
F	GF	7
H	GH	8
E	AE	12

2. **Minimum Spanning Trees and Subgraphs.** Let T be an MST of graph G. Given a connected subgraph H of G, show that $T \cap H$ is contained in some MST of H.

Solution: Let $T \cap H = \{e_1, \ldots, e_k\}$. We use the cut property repeatedly to show that there exists an MST of H containing $T \cap H$.

Suppose for $i < k$, $X = \{e_1, \ldots, e_i\}$ is contained in some MST of H. Removing the edge e_{i+1} from T divides T in two parts giving a cut $(S, G \backslash S)$ in G and a corresponding cut $(S_1, H \backslash S_1)$ of H with $S_1 = S \cap H$. Now, e_{i+1} must be the lightest edge in G (and hence also in H) crossing both cuts. If this were not true, we could include the lightest edge and remove e_{i+1}, creating a tree lighter than T and contradicting our assumption that T is an MST. Also, no other edges in T, and hence also in X, cross this cut, since T is a tree. We can then apply the cut property to get that $X \cup e_{i+1}$ must be contained in some MST of H. Continuing in this manner, we get the result for $T \cap H = \{e_1, \ldots, e_k\}$.

3. Edge Weight Incrementing. Consider an undirected graph $G = (V, E)$ with nonnegative edge weights $w_e \geq 0$. Suppose that you have computed a minimum spanning tree of G, and that you have also computed shortest paths to all nodes from a particular node $s \in V$.
Now suppose each edge weight is increased by 1: the new weights are $w'_e = w_e + 1$.

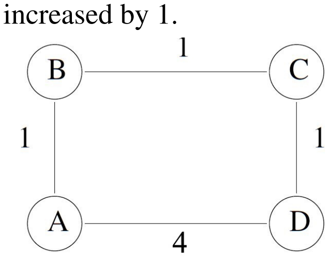
(a) Does the minimum spanning tree change? Give an example where it changes or prove it cannot change.

(b) Do the shortest paths change? Give an example where they change or prove they cannot change.

Solution:

(a) The minimum spanning tree does not change. Since, each spanning tree contains exactly $n-1$ edges, the cost of each tree is increased $n-1$ and hence the minimum is unchanged.

(b) The shortest paths may change because the number of edges in two competing shortest paths may be different. In the following graph, the shortest path from A to D changes from $AB-BC-CD$ to AD if each edge weight is increased by 1.



4. Minimum Spanning Trees

(a) Given an undirected graph $G = (V, E)$ and a set $E' \subset E$, briefly describe how to update Kruskal's algorithm to find the minimum spanning tree that includes all edges from E'.

(b) Assume you are given a graph $G = (V, E)$ with positive and negative edge weights and an algorithm that can return a minimum spanning tree when given a graph with only positive edges. Describe a way to transform G into a new graph G' containing only positive edge weights so that the minimum spanning tree of G can be easily found from the minimum spanning tree of G'.

Answer:

(a) Assuming E' doesn't have a cycle, add all edges from E' to the MST first, then sort $E \backslash E'$ and run Kruskal's as normal.

(b) We create G' by adding a large positive integer M to all the edge weights of G so that each edge weight is positive. The minimum spanning tree of G' is then the same as the minimum spanning tree of G.

Unlike Dijkstra's algorithm, which is finding minimum paths which may have different numbers of edges, all spanning trees of G must have precisely $|V|-1$ edges, conserving the MST. So, if the minimum spanning tree of G has weight w, the minimum spanning tree of G' has weight $w + (|V|-1)M$.