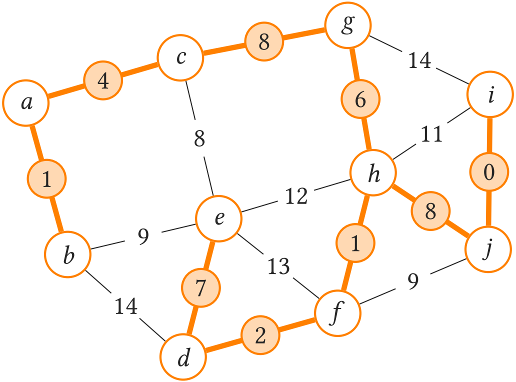
Minimum Spanning Trees

Minimum Spanning Tree

Minimum Spanning Tree

A *minimum spanning tree* (MST) of a connected, undirected, weighted graph G is a tree T with the minimum total weighting for its edges such that T contains all vertices of G.



Minimum Spanning Tree

Terminology

- *Spanning Subgraph.* A subgraph that contains all vertices, however, not necessarily all edges.
- *Spanning Tree.* A spanning subgraph that is a tree.
- *Minimum Spanning Tree.* A spanning tree of a weighted graph with minimum total edge weight.

Observation

- There can be several MSTs of the same weight.
- If all the edge weights of a given graph are equal, then every spanning tree of that graph is a MST.
- If each edge has a distinct weight, then there is only one MST.
  (We will see this later.)

Lemma

For any cycle C in a graph, if the weight of an edge e of C is strictly larger than the weights of all other edges of C, then this edge does not belong to any MST.

Proof

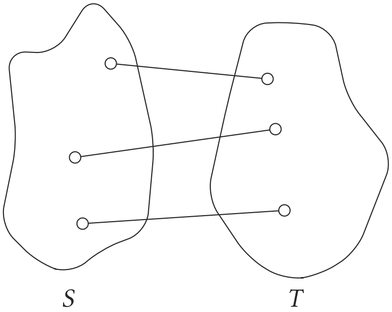
- Let uv be the edge of C with the largest weight, i. e., for all edges xy of C, $\omega(uv) > \omega(xy)$.
- Assume there is an MST T that contains uv.
- Remove uv from T (only the edge, not the vertices). Let T_u and T_v be the resulting subtrees.
- Then, there is an edge xy of C with $x \in T_u$ and $y \in T_v$.
- Because $\omega(uv) > \omega(xy)$, the total weight of $T_{xy} = T_u + T_v + xy$ is less than the total weight of T.
- This contradicts with T being an MST. $\square$

Cut

Cut

In a graph $G = (V, E)$, a set of edges C is a *cut* if there is a partition of V into S and T such that

$$C = \{ st \mid st \in E, s \in S, t \in T \}.$$



Lemma

For any cut C in the graph, if the weight of an edge e of C is strictly smaller than the weights of all other edges of C, then this edge belongs to all MSTs of the graph.

Proof

- Let uv be the edge in C with the strictly smallest weight, i. e., for all edges $xy \in C$, $\omega(uv) < \omega(xy)$.
- Assume there is an MST T that does not contain uv.
- Consider the path P from u to v in T.
- P contains an edge $xy \in C$.
- Since $\omega(uv) < \omega(xy)$, the total weight of the tree $T_{xy} = T_u v + xy$ is strictly less that the total weight of T.
- This contradicts with T being an MST. □

Kruskal’s Algorithm

Kruskal's Algorithm

Idea

- Assume we have a spanning forest F (set of trees) of G.
- Find edge uv with lowest weight that connects two distinct trees of F.
- Add uv to F, i. e., combine both trees into one.
- Repeat until the tree only contains one tree.

Runtime

- $\mathcal{O}(|E| \log |E|)$

Prim's Algorithm

Prim's Algorithm

Idea

- Initialize a tree with a single vertex, chosen arbitrarily from the graph.
- Of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and add it to the tree.
- Repeat until all vertices are in the tree.

Runtime

- $\mathcal{O}(|E| \log |V|)$ with binary heaps
- $\mathcal{O}(|E| + |V| \log |V|)$ with Fibonacci heaps.