# Notes, 6(c)

ECE 606

## Minimum Spanning Tree

Given a connected undirected graph $G = \langle V, E \rangle$, a *spanning tree* for it is a subgraph $T = \langle V, E_T \rangle$ that is a tree. Observe that $T$ contains all the vertices in $G$.

An efficient algorithm to construct a spanning tree: BFS.

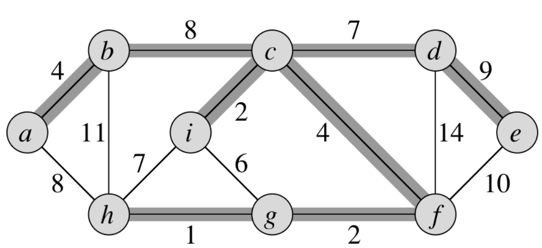
If $G = \langle V, E, w \rangle$ is a weighted, connected undirected graph, and $T$ is a spanning tree for $G$, then define the weight of $T = \langle V, E_T \rangle$ as:

$$w(T) = \sum_{\langle u,v \rangle \in E_T} w(u, v)$$

Given a weighted, connected undirected graph $G = \langle V, E, w \rangle$, a *minimum spanning tree* (MST) of $G$ is a spanning tree $T$ of $G$ that minimizes $w(T)$ across all spanning trees of $G$.

Minimum spanning tree problem:

- Input: weighted, undirected, connected $G = \langle V, E, w \rangle$.
- Output: $\pi[u]$ for every $u \in V$ that corresponds to an MST for $G$.



The notion of a *safe* edge:

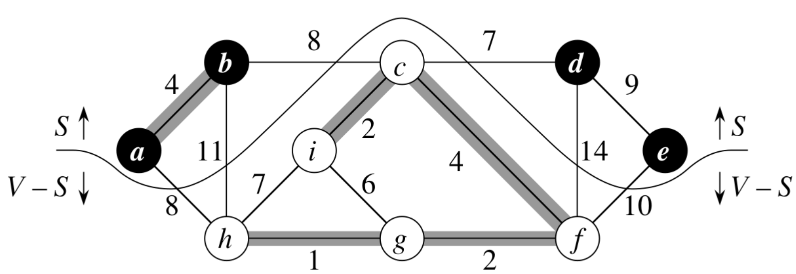
- If $A$ is a set of edges that is a subset of some MST of $G$, we say that an edge $\langle u, v\rangle$ is *safe* for $A$ if $A \cup \{\langle u, v\rangle\}$ is also a subset of some MST.

```
GENERIC-MST(G, w)
A ← ∅
while A does not form a spanning tree
    do find an edge (u, v) that is safe for A
        A ← A ∪ {(u, v)}
return A
```

Notions of a *cut*, and a *light edge that crosses a cut*:

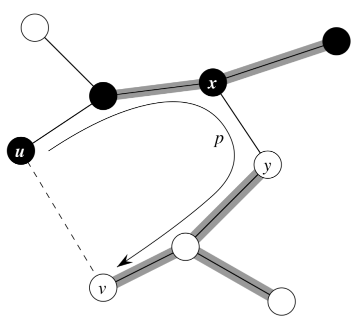
- A *cut* $\langle S, V \setminus S\rangle$ of an undirected $G = \langle V, E\rangle$ is a partition of $V$.
- An edge $\langle u, v\rangle \in E$ *crosses* the cut $\langle S, V \setminus S\rangle$ if $u \in S$ and $v \in V \setminus S$ (or vice versa).
- We say that a cut *respects* a set $A$ of edges if no edge in $A$ crosses the cut.
- An edge is a *light edge* that crosses a cut if its weight is the minimum of any edge that crosses the cut.

**Claim 1.** *Let $G = \langle V, E, w\rangle$ be a weighted connected undirected graph. Let $A \subseteq E$ be a subset of some MST of G and let the cut $\langle S, V \setminus S\rangle$ respect A. Then, a light edge $\langle u, v\rangle$ that crosses the cut $\langle S, V \setminus S\rangle$ is safe for A.*



*Proof.* Suppose $T$ is some MST that contains all the edges in $A$. We first observe that $\langle u, v \rangle \notin A$ because the cut $\langle S, V \setminus S \rangle$ respects $A$ and $\langle u, v \rangle$ crosses the cut.

If $\langle u, v \rangle \in T$, then we are done — $\langle u, v \rangle$ is safe for $A$. Otherwise, we need to prove that there exists some $T'$ such that: (i) $T'$ is an MST of $G$, (ii) $A \subseteq T'$, and, (iii) $\langle u, v \rangle \in T'$. We do this by construction. Specifically, we identify an edge $\langle x, y \rangle \in T$ as we discuss below, and adopt $T' = T \cup \{\langle u, v \rangle\} \setminus \{\langle x, y \rangle\}$ and prove that $T', A$ and $\langle u, v \rangle$ satisfy (i)–(iii).




We identify the edge $\langle x, y \rangle$ we remove from $T$ as follows. In $T$, there is a unique simple path $p$ such that $u \overset{p}{\rightsquigarrow} v$. As $u$ and $v$ are on different sides of the cut $\langle S, V \setminus S \rangle$, there must exist at least one edge in $p$ that also crosses the cut. We choose one of those edges as our $\langle x, y \rangle$.

Now we need to prove that we meet (i)–(iii) above. (i) Towards proving (i), we first we consider the weight of $T'$. Because $\langle u, v \rangle$ is a light edge that crosses $\langle S, V \setminus S \rangle$, $w(u, v) \leq w(x, y)$. Therefore, $w(T') \leq w(T)$. So provided $T'$ is a spanning tree, it is an MST.

To prove that $T'$ is a spanning tree, we first observe what the situation in $T$ is once we remove the edge $\langle x, y \rangle$. The set of vertices in $G$, $V$, can be partitioned in to two sets: $V_x$ is the set of vertices to which there exists a (simple) path in $T \setminus \{\langle x, y \rangle\}$ to $x$ and not to $y$, and $V_y$ is the set of vertices to which there exists a path to $y$ but not to x. Without loss of generality, assume that $u \in V_x, v \in V_y$. This means that every vertex in $V_x$ has a path to $u$ but not $v$, and every vertex in $V_y$ has a path to $v$ but not $u$. When we add the edge $\langle u, v \rangle$, every vertex in $G$ can reach every other vertex, and there are $|V| - 1$ edges in $T'$, and therefore $T'$ is a spanning tree.

(ii) To prove (ii), we observe that $\langle x, y\rangle \notin A$. Because the cut $\langle S, V \setminus S\rangle$ respects $A$ and $\langle x, y\rangle$ crosses the cut. Thus, $A \subseteq T'$, because $A \subseteq T$ and the only edge from $T$ not in $T'$ is $\langle x, y\rangle$.

(iii) $\langle u, v\rangle \in T'$ from the manner in which we construct $T'$ from $T$. $\square$

So, our algorithm is now clearer: start with $A = \emptyset$. Repeatedly identify a cut $\langle S, V \setminus S\rangle$ that respects $A$, then identify a light edge that crosses the cut and greedily add it to $A$. To identify a cut $\langle S, V \setminus S\rangle$ that respects $A$, we can leverage the following claim.

**Claim 2.** *Let $G = \langle V, E, w\rangle$ be a weighted connected undirected graph. Let $A \subseteq E$ be included in some MST for $G$, and let $C = \langle V_C, E_C\rangle$ be a connected component (tree) in the forest $G_A = \langle V, A\rangle$. If $\langle u, v\rangle$ is a light edge that connects $C$ to some other component in $G_A$, then $\langle u, v\rangle$ is safe for $A$.*

As a simple example, suppose $A = \emptyset$. Then, we can pick any $u \in V$, and set $C = \langle\{u\}, \emptyset\rangle$. Then, we would greedily pick $v \in V$ such that $\langle u, v\rangle \in E$ and $w(u, v)$ is the minimum across all edges that leave $u$.

Two algorithms that instantiate Generic-MST by exploiting the above two claims: Kruskal and Prim.

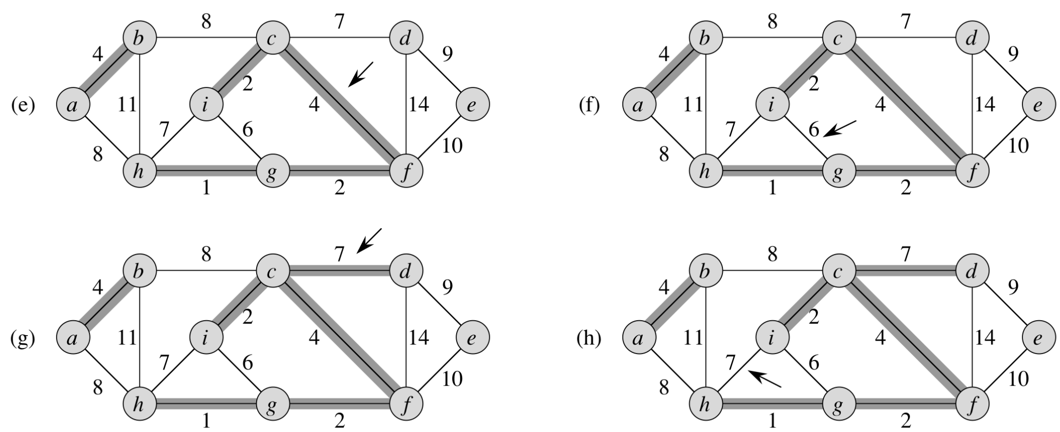
$\text{KRUSKAL}(G = \langle V, E, w \rangle)$

$A \leftarrow \emptyset$

**foreach** $\langle u, v \rangle \in E$ *in order of non-decreasing weight* **do**

  **if** $u$ *and* $v$ *are not already connected in* $A$ **then**

    $A \leftarrow A \cup \{\langle u, v \rangle\}$



PRIM is exactly DIJKSTRA with some vertex as the starting vertex, and a tweak to the relaxation routine.

RELAX$_{\mathrm{MST}}(u, v, w)$

**if** $v.d > w(u, v)$ **then**

  $v.d \leftarrow w(u, v)$

  $v.\pi \leftarrow u$