Chapter 11

Minimum Spanning Tree Recitation.

11.1 The Spanning Tree Problem.

Definition 11.1.1. A spanning tree T of a graph $G = (V, E)$ is a subset of edges in E such that T is a tree (having no cycles), and the graph (V, T) is connected.

Problem 11.1.1 (MST). Let $G = (V, E)$ be a weighted graph with weight function $w : E \to \mathbb{R}$. We extend the weight function to subsets of E by defining the weight of $X \subset E$ to be $w(X) = \sum_{e \in X} w(e)$. The minimum spanning tree (MST) of G is the spanning tree of G that has the minimal weight according to w. Note that in general, there might be more than one MST for G.

Definition 11.1.2. Let $U \subset V$. We define the cut associated with U as the set of outer edges of U, namely all the edges $(u, v) \in E$ such that $u \in U$ and $v \notin U$. We use the notation $X = (U, \bar{U})$ to represent the cut. We say that $E' \subset E$ respects the cut if $E' \cap X = \emptyset$.

Lemma 11.1.1 (The Cut-Lemma). *Let T be an MST of G. Consider a forest $F \subset T$ and a cut X that respects X (i.e. $F \cap X = \emptyset$). Then $F \cup \arg\min_e w(e)$ is also contained in some MST. Note that it does not necessarily have to be the same tree T.*

Proof. If $e \in T$ then $F \cup \{e\} \subset T$ and we done. So consider the second case $e \notin T \Rightarrow T \cup \{e\}$ has $|V|$ edges and therefore must has a cycle. Denote $\Gamma = T \cup \{e\}$. Let x, y be the ends of e (namely $e = (x, y)$). Denote the vertices subset defying the cut X by U. Since T is connected, there is a path $x \rightsquigarrow y$ in T denote it by $\mathcal{P}$, In addition, because $e \notin T \Rightarrow e \notin \mathcal{P}$ we have that there is must to be other edge in $\mathcal{P}$ connecting a vertex in U to a vertex in $\bar{U}$ [1]. Let e' be that edge.

The intersection $T \cap X$ must not be empty, otherwise there is no path connecting vertices on the opposite ends of the cut [2], and that is contradiction for T be a spanning tree. Denote by $e' \in T \cap X$ [3]. And consider $T' = T / \{e'\} \cup \{e\}$.

Our goal is to prove that T' is a minimum spanning tree, let's start with showing that T' is a spanning tree, Since it has $|V| - 1$ edges it's enough to show

[1] Otherwise walking a long $\mathcal{P}$ can not takes one out from U in contradiction for $\mathcal{P}$ leads to v

[2] In notations of Definition 11.1.2, no path connects vertices in U to vertices in $\bar{U}$

[3] If F spans either U or $\bar{U}$ then there is only one such e'. A good exercise would be prove it.

that T' is connected. Assume that T' is not a spanning tree, since T' has $|V| - 1$ edges, T must have cycle (otherwise T' is a tree with $|V|$ vertices $\Rightarrow T'$ is a spanning tree). Denote by u and v the ends vertices of e anf by x, y the ends of e' (i.e $e = (u, v), e' = (x, y)$). Without loose of generality assume that $u, x \in U$ and $v, y \in \bar{U}$, since T is connected ☐

In the lecture, you have seen the Kruskal algorithm for finding an MST. This algorithm constructs the MST iteratively, where in each step it holds a forest F contained in the MST and then looks for the minimal edge in a cut that it respects. Note that since F has no cycles, any edge $e \subset E$ that does not close a cycle in F must belong to some cut X that is respected by F. By enforcing the order of edges being examined to be increasing in weight, it follows that the first edge that does not close a cycle is also the one with the minimum weight among them. Therefore, by Lemma 11.1.1, we can conclude that the forest obtained by adding e into F is contained in the MST, and we can continue with it.

Result: Returns MST of given $G = (V, E, w)$

sorts the E according to w
define $F_0 = \emptyset$ and $i \leftarrow 0$
for $e \in E$ *in sorted order* **do**
    if $F_i \cup \{e\}$ *has no cycle* **then**
        $F_{i+1} \leftarrow F_i \cup \{e\}$
        $i \leftarrow i + 1$
    end
end
return F_i

Algorithm 1: Kruskal alg.

Claim 11.1.1. *Let G be a connected graph containing a cycle C. Then the subtraction of any an edge in C gives a connected graph.*

Proof. Assume, by contradiction, that a graph $G' = G/\{e\}$, where $e \in C$, is not connected. This means that there are two vertices u and v that have a path between them in G, but no such path exists in G'. Denote this path by $\mathcal{P}$ and observe that $e \in \mathcal{P}$, otherwise, $\mathcal{P}$ would also be a path from u to v in G'.

Denote the ends of e by $(x, y) = e$. Also, denote C by $\langle x_0, x_1, ..x_i, x, y, y_0, .., y_j \rangle$, where $y_j = x_0$ and there is an inequality for any other pair of vertices (we used the cycle definition). Then, there is a path $x \rightsquigarrow y$ in C, defined by

$$\langle x_i, x_{i-1}, .., x_1, x_0, y_{j-1}, y_{j-2}, .., y_0, y \rangle$$

We denote this path by $\mathcal{P}'$. By replacing e in $\mathcal{P}$ with $\mathcal{P}'$, we obtain a path $u \rightsquigarrow x \rightsquigarrow^{\mathcal{P}'} y \rightsquigarrow v$, which is a path between u and v that does not contain e. This contradicts the assumption that there is no path between u and v in G'. ☐