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 $e \notin T$. The intersection $T \cap X$ must not be empty, otherwise there is no path connecting vertices on the opposite ends of the cut (In notations of Definition 11.1.2, no path connects vertices in U to vertices in $\bar{U}$), and that is contradiction for T be a spanning tree. Denote by $e' \in T \cap X$ [1]. 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. 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, v the ends vertices of e (i.e $(u, v) = e$) . □

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.

[1] There is only one such e' but we haven't proved it yet.

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.