

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 $w : E \rightarrow \mathbb{R}$. 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 are 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 cut¹, and that is a contradiction for T to be a spanning tree. Denote by $e' \in T \cap X$ ². 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 a cycle (otherwise T' is a tree with $|V|$ vertices $\Rightarrow T'$ is a spanning tree). Denote by u and v the end vertices of e (i.e. $(u, v) = e$). \square

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.

¹In notations of Definition 11.1.2, no path connects vertices in U to vertices in \bar{U}

²If F spans either U or \bar{U} then there is only one such e' . A good exercise would be to prove it.

Note that since F has no cycles, any edge $e \in E$ 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 $G = (V, E, w)$

```

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

```

Algorithm 1: Kruskal alg.

Claim 11.1.1. Let G be a connected graph containing a cycle C . Then subtracting an edge of C gives a connected graph.

Proof. Assume trough contradiction that a graph $G' = G/\{e\}$, where $e \in C$ is not connected. Means that there are two vertices u and v that has a path between them in G but, no such a path in G' denote that path by \mathcal{P} and observes that $e \in \mathcal{P}$ otherwise \mathcal{P} would be also $u \rightsquigarrow v$ path in G' .

Denote the ends of e by $(x, y) = e$. Also, denote C by $\langle x_0, x_1, \dots, x_i, x, y, y_0, \dots, y_j \rangle$, 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

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

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' . \square