

## 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 cut (In notations of Definition 11.1.2, no path connects vertices in  $U$  to vertices in  $\bar{U}$ ), and that is a contradiction for  $T$  to be a spanning tree. Denote by  $e' = T \cap X$ . 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 it is a tree with  $|V|$  vertices  $\Rightarrow$  a spanning tree). Denote by  $u, v$  the ends 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. 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.