Summary on Lecture 24, March 4, 2016

Optimal spanning trees: Prim's Algorithm in more detail

For a given finite connected graph $G = (V(g), E(G))$, we are looking for a spanning tree $T \subset G$ of minimal weight.

Recall Prim's algorithm:

Prim's Algorithm$(G = (V(G), E(G)),\ \mathtt{wt} : E(G) \to (0, \infty))$
`Input:` A finite weighted connected graph $(G, \mathtt{wt})$ with edges listed in any order
`Output:` A set E of edges of an optimal spanning tree for G)

```
Set E = ∅.  Choose w in V(G) and set V := {w}.
while V = V(G) do
   Choose an edge {u,v} in E(G) of smallest possible weight
      with u ∈ V and v ∈ V(G) \ V.
   Put {u,v} in E and put v in V.
return E
```

Theorem. *Prim's algorithm produces an optimal spanning tree for a connected weighted graph.*

Proof. Theorem 1 and the way the algorithm **Tree** works, show that the graph the Prim's algorithm is producing is indeed a spanning tree. We have to show that it is an optimal one. We consider the statment

$$\mathbf{S} := \text{``The graph } T \text{ is contained in an optimal spanning tree of } G$$

It holds at the beginning since T is a single vertex. We claim that **S** is an invariant of the while loop. Suppose that, at the beginning of some pass through the while loop, T is contained in the minimum spanning tree T^* of G. Suppose that the algorithm now chooses the edge $\{u, v\}$. If $\{u, v\} \in E(T^*)$, then the new T is still contained in T^*, which is wonderful. Suppose not. Because T^* is a spanning tree, there is a path in T^* from u to v. Since $u \in V$ and $v \notin V$, there must be some edge in the path that joins a vertex z in V to a vertex $w \in V(G) \setminus V$.

Since Prim's algorithm chose $\{u, v\}$ instead of $\{z, w\}$, we have $\mathtt{wt}\{u, v\} \leq \mathtt{wt}\{z, w\}$. Take the edge $\{z, w\}$ out of $E(T^*)$ and replace it with $\{u, v\}$. The new graph T^{**} is still connected, so it's a tree. Since $W(T^{**}) \leq W(T^*)$, the graph T^{**} is also an optimal spanning tree, and T^{**} contains the new T. At the end of the loop, T is still contained in some optimal spanning tree, as we wanted to show. □