## CSC373— Algorithm Design, Analysis, and Complexity — Spring 2018

## Tutorial Exercise 1: Minimum Spanning Trees

---

The first two questions gives you practice proving statements about trees. (And it might help with Assignment 1.) The third question simply involves reading and understanding a detailed proof of Prim's algorithm.

Bring any questions you may have to the next tutorial where you will get a chance to discuss these with TAs and other student groups.

**1. (Part 1) Proving Statements about Trees.** Let $G = (V, E)$ be a tree (and therefore an undirected graph). Here we look at careful proofs about the effect of removing one edge.

Let $e = (e_1, e_2) \in E$ and consider the graph formed by deleting $e$ from $G$, i.e., the graph $G_e \equiv (V, E\backslash\{e\})$. We wish to carefully prove that, in this case, $G_e$ is a forest consisting of two trees, $(V_1, E_1)$ and $(V_2, E_2)$, with $V_1 \cap V_2 = \emptyset$ and with $V_1 \cup V_2 = V$. We do this in the following steps.

**1a)** Intuitively, removing the edge $e = (e_1, e_2)$ splits the tree into two connected parts, one of which would contains $e_1$ and the other $e_2$. Suppose we define $V_1 = \{v \in V \mid v \text{ is reachable from } e_1 \text{ in } G_e\}$, and define $V_2$ similarly, as the set reachable from $e_2$. In order to show there are exactly two connected parts, prove $V_1 \cup V_2 = V$ for this definition of $V_1$ and $V_2$.

**1b)** Next prove $V_1 \cap V_2 = \emptyset$.

**1c)** Prove $(V_k, E_k)$ is acyclic for $k = 1, 2$.

**1d)** Prove $(V_k, E_k)$ is connected for $k = 1, 2$.

**2. (Part 2) Proving Statements about Trees.** Let $G = (V, E)$ be an undirected graph and suppose $(V_1, E_1)$ and $(V_2, E_2)$ are subgraphs that are trees. We assume that $V_1 \cap V_2 = \emptyset$. Here we look at adding one edge, $e = (e_1, e_2) \in E$, with $e_1 \in V_1$ and $e_2 \in V_2$ to form a new subgraph $(V_1 \cup V_2, E_1 \cup E_2 \cup \{e\})$.

Carefully prove that, in this case, the subgraph $T \equiv (V_1 \cup V_2, E_1 \cup E_2 \cup \{e\})$ is a tree.

**2a)** First prove $T$ is a subgraph of $G$.

**2b)** Prove $T$ is connected.

**2c)** Prove $T$ has $|V_1| + |V_2|$ vertices and $|V_1| + |V_2| - 1$ edges.

**(Sketch of Soln 2c)** The number of vertices in $T$ is $|V_1| + |V_2|$ (since $V_1 \cap V_2 = \emptyset$). Similarly, the number of edges in $T$ is $|E_1| + |E_2| + 1$. However

$$\begin{aligned} |E_1| + |E_2| + 1 &= (|V_1| - 1) + (|V_2| - 1) + 1, \quad \text{since } (V_k, E_k) \text{ are trees} \\ &= |V_1| + |V_2| - 1, \end{aligned}$$

which is the desired result.

**2d)** Prove that $T$ is a tree.

**3. Read a proof that Prim's algorithm generates an MST.**

Your task is simply to read and understand the following proof.

```
A proof of Prim's algorithm showing that each step "is promising".

Theorem: Given a connected undirected graph G = (V, E), with
```

edge weights w(e), Prim's algorithm constructs a MST of G.

Proof: We use the loop invariant:

L(k): Let T_k = (S_k, F_k) be the subgraph that Prim has constructed
with k = |S_k| (after the execution of the loop (k-1) times). Then
there exists an MST T=(V,F) of G=(V, E) s.t. T_k is a subgraph of T.

That is, T_k "is promising".

We prove L(k) is true for k = 1 to |V| using induction.

For the base case, k=1, T_1 = ({v}, {}). Given the
existence of an MST T for G, T_1 must be a subgraph of T.
We leave the existence of an MST for G to the reader.

Let 1< k <= |V|.

Assume L(k-1) is true. We need to prove L(k) is true.

Since L(k-1) is assumed true, let T_(k-1) = (S_(k-1), F_(k-1)) be
the subgraph generated during the execution of the algorithm
with |S_(k-1)| = k-1. And let T be an MST of G which contains T_(k-1).
(Such a T exists according to L(k-1).)

Let e = (u, v), with u in S_(k-1) and v in V \ S_{k-1}, be
the k-th edge added by Prim.

If edge e is in the MST T = (V, F), then L(k) follows.
So we are left with considering the case e is not in F.

Suppose e = (u,v) is not in F. Since T is a spanning tree,
there exists a simple path P = (u, a1, a2, ... an, v) in T,
from u to v. Let f = (a,b) = (a(i), a(i+1)) be the first edge
on this path that has a(i) in S_(k-1) and a(i+1) in V\S_(k-1).
Note u in S_(k-1) and v is in the complement, so such an edge
must exist. Also, all the edges in P are in T and e
is not in T, so f does not equal e.

Claim: The graph Tp = (V, (F\{f})U{e})) is a tree.
Pf Omitted: Left to reader to show: 1) Tp has the right number
of edges, namely |V|-1, and 2) Tp is connected.
Together these imply Tp is a tree (Property 2 in the MST slides.)
Hint: To show Tp is connected, let x and y be any elements in V, and
construct a x-y path in Tp. To do this use part of the simple cycle
C = (u, a1, a2, ... an, v, u) in the graph A=(V, F U {e}),
where P = (u, a1, ... an, v) is as constructed above.

Finally consider the weight of this tree Tp,
(*) w(Tp) = w(T) - w(f) + w(e)

There are three cases:

a) w(f) < w(e): Impossible, since Prim's alg would not have chosen e in the presence of the edge , for which f = (a, b) with w(f) < w(e), a is in S_(k-1), and b in V\S_(k-1).

b) w(f) = w(e): Then (*) implies w(Tp) = w(T) and, since we know T is a MST, then Tp must be an MST too. Therefore L(k) holds with this modified MST Tp.

c) w(f) > w(e): Impossible, since in this case (*) implies w(Tp) < w(T), but this contradicts T being a MST.

Therefore L(k) must be true.

It follows by induction that L(k) is true for k = 1 to |V|.

Finally, L(|V|) implies that the computed subgraph is an MST for G.

=====

Note: At each stage of Prim's algorithm T_k is a subtree. We carefully didn't make use of this above (and we didn't prove it). Because of this, the same loop invariant (with "Prim" replaced by Kruskal) applies to Kruskal's algorithm. The proof of Kruskal's algorithm then requires only a small modification.