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

Suggested Solutions for Tutorial Exercise 2: MST

---

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.

(Solution 1a) Pf: Let v be any vertex in V and consider the unique simple path $P(e_1, v)$ in G (since G is a tree). Here e_1 is as above, a vertex of edge e. We will consider three cases for such a path. First, let's get rid of a nuisance case, say case 1, where the number of edges in $P(e_1, v)$ is zero. So in this case, $v = e_1 \in V_1$..

Otherwise in case 2, say, we can assume there is at least one edge in $P(e_1, v)$. So $P(e_1, v) = (e_1, p_2, \ldots, p_n)$ with $n \geq 2$ and $p_n = v$. There are now two subcases. Either (Case 2a) $p_2 = e_2$ (i.e., the first edge on $P(e_1, v)$ is e) or (Case 2b) $p_2 \neq e_2$. Note that these cases are exhaustive.

In case 2a we have $P(e_1, v)$ is simple and the second vertex in the path is $p_2 = e_2$. So the first edge on the path $P(e_1, v)$ is e. Since P is simple, it cannot revisit either e_1 or e_2 after this first edge. Therefore the endpoint v must be reachable from e_2 by only using edges in $E\backslash\{e\}$. That is, $v \in V_2$.

In case 2b, we have $p_2 \neq e_2$. Again, using the property that $P(e_1, v)$ is simple, it must be the case that e_1 does not appear anywhere on P except at that first vertex. Therefore, the edge (e_1, e_2), cannot appear on $P(e_1, v)$ in either order. That is, v is reachable from e_1 without using the edge e, so $v \in V_1$.

Therefore, in all cases, we have $v \in V_1 \cup V_2$. Since v was an arbitrary element of V we have shown $V \subset V_1 \cup V_2$. However, by construction, V_1 and V_2 must be subsets of V, and so we have $V = V_1 \cup V_2$. ■

(Sketch of Soln 1a) Let v be any vertex in V. Since G is a tree, there exists a simple path, $P(e_1, v)$, in G. Here e_1 is one of the vertices in the edge (e_1, e_2) that we remove. Since the path P is simple, there are two cases, either e is the first edge on the path (and never appears on P again, in either order) or e is not on P at all (in either order). In the first case, observe that after traversing the first edge (e_1, e_2) of P, the remainder of P shows v is reachable from e_2 in G_e. Moreover, in the second case, it follows that v is reachable from e_1 in G_e. Therefore, v must be in either V_1 or V_2.

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

(Solution 1b) Pf: By contradiction. Suppose $x \in V_1 \cap V_2$. Then by the definition of V_1 and V_2 there must be paths from e_k to x, say $P(e_k, x)$ for $k = 1, 2$, which do not use the edge e. But this implies that G contains the cycle $C = \text{cat}(P(e_2, x), \text{reverse}(P(e_1, x)), (e_1, e_2))$. Here, $\text{reverse}(P(e_1, x))$ is the path $P(e_1, x)$ in reverse order, which must be in G_e since all the edges are undirected. Note, by construction, this cycle C contains e exactly once, so it is not trivial. But this contradicts G being a tree. So there can be no element $x \in V_1 \cap V_2$.
■

(Sketch of Soln 1b) Use contradiction. Suppose $x \in V_1 \cap V_2$. Then, by the definitions of V_k, $k = 1, 2$, there exist simple paths $P(e_1, x)$ and $P(e_2, x)$ in G_e. Note that e cannot be on either path, so $C = \text{cat}(P(e_1, x), \text{reverse}(P(e_2, x)), (e_2, e_1))$ is a cycle which contains e exactly once. So C must contain a simple cycle, which contradicts G being a tree.

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

(Solution 1c) Pf: Note we haven't yet defined E_k, but the following is the only choice which makes sense. For $k = 1, 2$, define $E_k = \{(u, v) \in E \mid \text{ both } u \text{ and } v \text{ are in } V_k\}$. Then, by construction, (V_k, E_k), for $k = 1, 2$, are subgraphs of both G and G_e.

We show (V_k, E_k) is acyclic using contradiction. Suppose there is a simple cycle in at least one of these two subgraphs. Without loss of generality (WLOG) we can assume it is for $k = 1$. However, such a cycle in (V_1, E_1) must also be a cycle in $G = (V, E)$ (since (V_1, E_1) is a subgraph of G). But this contradicts the assumption that G is a tree and hence acyclic. ■

(Sketch of Soln 1c) Note, we need to define $E_k = \{(u, v) \in E \mid \text{ both } u \text{ and } v \text{ are in } V_k\}$. Since (V_k, E_k) is a sub-graph of G, any cycle in (V_k, E_k) would be a cycle in G, which would contradict G being a tree.

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

(Solution 1d) Pf: Let $k = 1$ or 2. Let u, v be any elements of V_k, with $u \neq v$. We will show u and v must be connected in the subgraph (V_k, E_k) by first constructing two paths, namely $P(e_k, u)$ and $P(e_k, v)$, that connect e_k to these two points u and v, where neither of these paths uses the edge e. By the definitions of V_k and E_k, both of these paths must exist in the subgraph (V_k, E_k). Then the reverse of $P(e_k, u)$, say $P(u, e_k)$, must also be in (V_k, E_k) (since E and E_k are undirected). Therefore the concatenation $\text{cat}(P(u, e_k), P(e_k, v))$ must be a $u - v$ path in the subgraph (V_k, E_k). Since u, v were any vertices in V_k, we have shown that (V_k, E_k) is connected. ■

(Sketch of Soln 1d) Recall (V_k, E_k) is a sub-graph of G_e, where V_k consists of all vertices reachable from the endpoint e_k using edges in E_k. For any $u, v \in V_k$, we therefore have each of u and v are connected to e_k and, by transitivity of connectedness, must be connected to each other within (V_k, E_k).

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.

(Solution 2a) Pf: By construction, we see the vertices of T satisfy $V_1 \cup V_2 \subset V$. Moreover, the edges satisfy $E_1 \cup E_2 \cup \{e\} \subset E$. What more do we need to argue that T is a subgraph? We just need to show that the endpoints of the edges in $E_1 \cup E_2 \cup \{e\}$ are all in $V_1 \cup V_2$. But, from above it follows the the endpoints of all edges in E_k are in V_k (since (V_k, E_k) is assumed to be a subgraph). The only additional edge is $e = (e_1, e_2)$, and $e_1 \in V_1$ while $e_2 \in V_2$, so the result follows. ■

(Sketch of Soln 2a) Clearly all the vertices and edges in T are in G. And all the endpoints of the edges in T are vertices in T, by construction, so T is a well-formed graph.

2b) Prove T is connected.

(Solution 2b) Pf: Let u, v be any two distinct vertices in $V_1 \cup V_2$. Let $k = 1$ or 2. If both u and v are in V_k then, by the connectivity of (V_k, E_k), there must be a (unique simple) path between u and v in (V_k, E_k). The only other case is when one of u and v is in V_1, while the other is in V_2.

WLOG assume $u \in V_1$ and $v \in V_2$. Since $e_1 \in V_1$ there must be a (unique simple) path $P(u, e_1)$ in the subgraph (V_1, E_1). Similarly, since $e_2 \in V_2$, there must be a path $P(e_2, v)$ in the subgraph (V_2, E_2). Therefore, we can construct a $u - v$ path in T by concatenating $P(u, e_1)$, (e_1, e_2) and $P(e_2, v)$.

Since u and v were any elements of $V_1 \cup V_2$, and since the above cases are exhaustive, we conclude that T is connected. ■

(Sketch of Soln 2b) By construction, every vertex in T is in V_1 or V_2 and is therefore connected to either e_1 or e_2. Moreover, these latter two vertices are connected by the edge e in T. Therefore, by transitivity, every vertex in T is connected to e_1, and so every pair of vertices in T must be connected.

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

(Solution 2c) Pf: Since (V_k, E_k) are both trees, $|E_k| = |V_k| - 1$ for $k = 1, 2$. Moreover, since $V_1 \cap V_2 = \emptyset$, the number of vertices in T is $|V_1| + |V_2|$. It also follows from $V_1 \cap V_2 = \emptyset$ that $E_1 \cap E_2 = \emptyset$ (i.e., there cannot be a common edge in E_1 and E_2 since the subgraphs use distinct sets of vertices V_1 and V_2). Note that $e \notin E_1 \cup E_2$. Therefore we have the number of edges in T is $|E_1 \cup E_2 \cup \{e\}| = |E_1| + |E_2| + 1$. From above, this equals $|V_1| - 1 + |V_2| - 1 + 1 = |V_1| + |V_2| - 1$. ■

(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.

(Solution 2d) Pf: This follows from Property 2 in the lecture notes, along with parts (2a, b, c) above. ■

(Sketch of Soln 2d) Same as above.