Problems for Recitation 8

1 Build-up error

Recall a graph is connected iff there is a path between every pair of its vertices.

False Claim. If every vertex in a graph has positive degree, then the graph is connected.

1. Prove that this Claim is indeed false by providing a counterexample.

2. Since the Claim is false, there must be a logical mistake in the following bogus proof. Pinpoint the first logical mistake (unjustified step) in the proof.

Proof. We prove the Claim above by induction. Let \( P(n) \) be the proposition that if every vertex in an \( n \)-vertex graph has positive degree, then the graph is connected.

Base cases: \((n \leq 2)\). In a graph with 1 vertex, that vertex cannot have positive degree, so \( P(1) \) holds vacuously.

\( P(2) \) holds because there is only one graph with two vertices of positive degree, namely, the graph with an edge between the vertices, and this graph is connected.

Inductive step: We must show that \( P(n) \) implies \( P(n + 1) \) for all \( n \geq 2 \). Consider an \( n \)-vertex graph in which every vertex has positive degree. By the assumption \( P(n) \), this graph is connected; that is, there is a path between every pair of vertices. Now we add one more vertex \( x \) to obtain an \((n + 1)\)-vertex graph:

All that remains is to check that there is a path from \( x \) to every other vertex \( z \). Since \( x \) has positive degree, there is an edge from \( x \) to some other vertex, \( y \). Thus, we can
obtain a path from $x$ to $z$ by going from $x$ to $y$ and then following the path from $y$ to $z$. This proves $P(n + 1)$.

By the principle of induction, $P(n)$ is true for all $n \geq 0$, which proves the Claim.

2 The Grow Algorithm

Yesterday in lecture, we saw the following algorithm for constructing a minimum-weight spanning tree (MST) from an edge-weighted $N$-vertex graph $G$.

ALG-GROW:

1. Label the edges of the graph $e_1, e_2, \ldots, e_t$ so that $wt(e_1) \leq wt(e_2) \leq \ldots \leq wt(e_t)$.
2. Let $S$ be the empty set.
3. For $i = 1 \ldots t$, if $S \cup \{e_i\}$ does not contain a cycle, then extend $S$ with the edge $e_i$.
4. Output $S$.

2.1 Analysis of ALG-GROW

In this problem you may assume the following lemma from the problem set:

Lemma 1. Suppose that $T = (V, E)$ is a simple, connected graph. Then $T$ is a tree iff $|E| = |V| - 1$.

In this exercise you will prove the following theorem.

Theorem. For any connected, weighted graph $G$, ALG-GROW produces an MST of $G$.

(a) Prove the following lemma.

Lemma 2. Let $T = (V, E)$ be a tree and let $e$ be an edge not in $E$. Then, $G = (V, E \cup \{e\})$ contains a cycle.

(Hint: Suppose $G$ does not contain a cycle. Is $G$ a tree?)

(b) Prove the following lemma.
Lemma 3. Let \( T = (V, E) \) be a spanning tree of \( G \) and let \( e \) be an edge not in \( E \). Then there exists an edge \( e' \neq e \) in \( E \) such that \( T^* = (V, E - \{e\} \cup \{e\}) \) is a spanning tree of \( G \).

(Hint: Adding \( e \) to \( E \) introduces a cycle in \((V, E \cup \{e\})\).)

(c) Prove the following lemma.

Lemma 4. Let \( T = (V, E) \) be a spanning tree of \( G \), let \( e \) be an edge not in \( E \) and let \( S \subseteq E \) such that \( S \cup \{e\} \) does not contain a cycle. Then there exists an edge \( e' \neq e \) in \( E - S \) such that \( T^* = (V, E - \{e\} \cup \{e\}) \) is a spanning tree of \( G \).

(Hint: Modify your proof to part (b). Of all possible edges \( e' \neq e \) that can be removed to construct \( T^* \), at least one is not in \( S \).)

(d) Prove the following lemma.

Lemma 5. Define \( S_m \) to be the set consisting of the first \( m \) edges selected by ALG-GROW from a connected graph \( G \). Let \( P(m) \) be the predicate that if \( m \leq |V| \) then \( S_m \subseteq E \) for some MST \( T = (V, E) \) of \( G \). Then \( \forall m . P(m) \).

(Hint: Use induction. There are two cases: \( m + 1 > |V| \) and \( m + 1 \leq |V| \). In the second case, there are two subcases.)

(e) Prove the theorem. (Hint: Lemma 5 says there exists an MST \( T = (V, E) \) for \( G \) such that \( S \subseteq E \). Use contradiction to rule out the case in which \( S \) is a proper subset of \( E \).)
MIT OpenCourseWare
http://ocw.mit.edu

6.042J / 18.062J Mathematics for Computer Science
Fall 2010

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.