Staff Solutions to In-Class Problems Week 9, Mon.

Staff Note: Trees, Ch. 11.11

Problem 1.
Let $G$ be a $4 \times 4$ grid with vertical and horizontal edges between neighboring vertices. Formally,

$$V(G) = [0, 3]^2 := \{(k, j) \mid 0 \leq k, j \leq 3\}.$$

Letting $h_{i,j}$ be the horizontal edge $((i, j)\rightarrow(i + 1, j))$ and $v_{j,i}$ be the vertical edge $((j, i)\rightarrow(j, i + 1))$ for $i \in [0, 2], j \in [0, 3]$. The weights of these edges are

$$w(h_{i,j}) := \frac{4i + j}{100},$$
$$w(v_{j,i}) := 1 + \frac{i + 4j}{100}.$$

(A picture of $G$ would help; you might like to draw one.)

(a) Construct a minimum weight spanning tree (MST) for $G$ by initially selecting the minimum weight edge, and then successively selecting the minimum weight edge that does not create a cycle with the previously selected edges. Stop when the selected edges form a spanning tree of $G$. (This is Kruskal’s MST algorithm.) Explain how the “gray edge” Lemma 11.10.11 justifies this algorithm.

Solution. The edges are in the order that they are constructed by the given algorithm.

Answer: $h_{0,0}h_{0,1}h_{0,2}h_{0,3}h_{1,0}h_{1,1}h_{1,2}h_{1,3}h_{2,0}h_{2,1}h_{2,2}h_{2,3}v_{0,0}v_{0,1}v_{0,2}$

From the text: An edge does not create a cycle if it connects different components. The edge chosen by Kruskal’s algorithm will be the minimum weight gray edge when the components it connects are assigned different colors.

(b) Grow an MST for $G$ starting with the tree consisting of the single vertex $(1, 2)$ and successively adding the minimum weight edge with exactly one endpoint in the tree. Stop when the tree spans $G$. (This is Prim’s MST algorithm.) Explain how the “gray edge” Lemma 11.10.11 justifies this algorithm.

Solution. Answer: $h_{0,2}h_{1,2}h_{2,2}v_{0,0}h_{0,1}h_{1,1}v_{0,0}h_{0,0}h_{1,0}h_{2,0}h_{2,0}v_{0,3}h_{1,3}h_{2,3}$

From the text: This is the algorithm that comes from coloring the growing tree white and all the vertices not in the tree black. Then the gray edges are the ones with exactly one endpoint in the tree.

(c) Grow an MST for $G$ by treating the vertices $(0, 0), (0, 3), (2, 3)$ as 1-vertex trees and then successively adding, for each tree in parallel, the minimum weight edge among the edges with one endpoint in the tree. Continue as long as there is no edge between two trees, then go back to applying the general gray edge method until the parallel trees merge to form a spanning tree of $G$. (This is 6.042’s parallel MST algorithm.)
Solution. Done in parallel:
T1@(0,0): \( h_{0,3} \) (merges with T2)
T2@(0,3): \( h_{1,3}, h_{3,2} h_{0,2} h_{1,2}, v_{0,1} \) (merges with T3)
T3@(2,3): \( h_{0,0} h_{1,0} h_{2,0} v_{0,0} h_{0,1} h_{1,1} h_{2,1} \)

(d) Verify that you got the same MST each time.

Solution. They are the same —if no mistake was made. Problem 11.50 explains why there is a unique MST for any finite connected weighted graph where no two edges have the same weight.

(e) Look up the proof of the “gray edge” Lemma 11.10.11, and spend up to 15 minutes drawing one or two figures that could be added to the text to help make the proof clearer.

Solution. When we get the figures, we’ll add them to the text.

Problem 2.
A procedure for connecting up a (possibly disconnected) simple graph and creating a spanning tree can be modelled as a state machine whose states are finite simple graphs. A state is final when no further transitions are possible. The transitions are determined by the following rules:

Procedure create-spanning-tree

1. If there is an edge \( u—v \) on a cycle, then delete \( u—v \).
2. If vertices \( u \) and \( v \) are not connected, then add the edge \( u—v \).

(a) Draw all the possible final states reachable starting with the graph with vertices \( \{1, 2, 3, 4\} \) and edges \( \{(1—2), (3—4)\} \).

Solution. It’s not possible to delete any edge. The procedure can only add an edge connecting exactly one of vertices 1 or 2 to exactly one of vertices 3 or 4, and then terminate. So there are four possible final states.

(b) Prove that if the machine reaches a final state, then the final state will be a tree on the vertices graph on which it started.

Solution. We use the characterization of a tree as an acyclic connected graph.
A final state must be connected, because otherwise there would be two unconnected vertices, and then a transition adding the edge between them would be possible, contradicting finality of the state.
A final state can’t have a cycle, because deleting any edge on the cycle would be a possible transition.

(c) For any graph, \( G’ \), let \( e \) be the number of edges in \( G’ \), \( c \) be the number of connected components it has, and \( s \) be the number of cycles. For each of the quantities below, indicate the strongest of the properties that it is guaranteed to satisfy, no matter what the starting graph is and be prepared to briefly explain your answer.

The choices for properties are: constant, strictly increasing, strictly decreasing, weakly increasing, weakly decreasing, none of these.
(i) $e$

Solution. none of these

(ii) $c$

Solution. weakly decreasing

(iii) $s$

Solution. weakly decreasing

(iv) $e - s$

Solution. weakly increasing

(v) $c + e$

Solution. weakly decreasing

(vi) $3c + 2e$

Solution. strictly decreasing

(vii) $c + s$

Solution. strictly decreasing

(d) Prove that one of the quantities from part (c) strictly decreases at each transition. Conclude that for every starting state, the machine will reach a final state.

Solution. If a value associated with states, known as a derived variable, is nonnegative integer-valued and decreases at each transition, then the machine must reach a final state after at most as many transitions as the initial value of the variable. So we need only identify such a derived variable. There are two in the list above, namely (vi) and (vii).

To show that the variable (vi) strictly decreases, note that the rule for deleting an edge ensures that the connectedness relation does not change, so neither does the number of connected components $c$. Meanwhile the number of edges $e$ decreases by one when an edge is deleted. Therefore the variable $3c + 2e$ decreases by 2. The rule for adding an edge ensures that the number of connected components $c$ decreases by one and the number of edges $e$ increases by one. Therefore the variable $3c + 2e$ decreases by 1.

To show that the variable (vii) strictly decreases, note that the rule for deleting an edge ensures that the number of connected components $c$ does not change and the number of cycles $s$ decreases by $n$, where $n \geq 1$. Therefore the variable $c + s$ decreases by $n$. The rule for adding an edge ensures that the number of connected components $c$ decreases by one and the number of cycles $s$ does not change. Therefore the variable $c + s$ decreases by one.

Problem 3.
Prove that a graph is a tree iff it has a unique path between every two vertices.
Figure 1  If there are two paths between $u$ and $v$, the graph must contain a cycle.

**STAFF NOTE:** Students should be told not to look up the proof in the text until they try this on their own.

**Solution.** Theorem 11.10.3.2 shows that in a tree there are unique paths between any two vertices.

**STAFF NOTE:** Since a tree is connected, there is at least one path between every pair of vertices. To show paths are unique:

**first proof:** Suppose for the purposes of contradiction, that there are two different paths between some pair of vertices. Then there are two distinct paths $p \neq q$ between two vertices with minimum total length $|p| + |q|$. If these paths shared a vertex, $w$, other than at the start and end of the paths, then the parts of $p$ and $q$ from start to $w$, or the parts of $p$ and $q$ from $w$ to the end, must be distinct paths between the same vertices with total length less than $|p| + |q|$, contradicting the minimality of this sum. Therefore, $p$ and $q$ are distinct paths with no vertices in common besides their endpoints, and so

$$p \sim \text{reverse}(q)$$

is a cycle.

**second proof:** Beginning at $u$, let $x$ be the first vertex where the paths diverge, and let $y$ be the next vertex they share. (For example, see Figure 1.) Then there are two paths from $x$ to $y$ with no common edges, which defines a cycle. This is a contradiction, since trees are acyclic. Therefore, there is exactly one path between every pair of vertices.

Conversely, suppose we have a graph, $G$, with unique paths. Now $G$ is connected since there is a path between any two vertices. So we need only show that $G$ has no cycles. But if there was a cycle in $G$, there are two paths between any two vertices on the cycle (going one way around the cycle or the other way around), a violation of uniqueness. So $G$ cannot have any cycles.

**Problem 4.**
Prove Corollary 11.10.12: If all edges in a finite weighted graph have distinct weights, then the graph has a unique MST.

**Hint:** Suppose $M$ and $N$ were different MST’s of the same graph. Let $e$ be the smallest edge in one and not the other, say $e \in M - N$, and observe that $N + e$ must have a cycle.

**Solution.** Assume for the sake of contradiction that $M$ and $N$ were different MST’s of the same graph. Let $e$ be a minimum weight edge as in the hint.

Since $N$ is a spanning tree, we know that $N + e$ is connected, and it has too many edges to be a tree, so $N + e$ has a cycle. Since $M$ has no cycles, the cycle in $N + e$ cannot consist solely of edges from $M$. So there must be an edge $g$ on the cycle that is larger than $e$. Removing $g$ from $N + e$ leaves a connected graph with the same number of nodes and edges as $N$, so $N + e - g$ must be a spanning tree. But $N + e - g$ weighs $w(g) - w(e)$ less than $N$, contradicting the minimality of $N$. 
