In-Class Problems Week 7, Wed.

Problem 1.
Let G be a 4×4 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 $\langle (i, j)—(i+1, j) \rangle$ and $v_{j,i}$ be the vertical edge $\langle (j, i)—(j, i+1) \rangle$ 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.)

For any step in Kruskal'sprocedure, describe a black-white coloring of the graph components so that the edge Kruskal chooses is the minimum weight "gray edge" according to Lemma **??**.

(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.) For any step in Prim's procedure, describe a black-white coloring of the graph components so that the edge Prim chooses is the minimum weight "gray edge" according to Lemma **??**.

(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.)

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

Problem 2.
Prove that a graph is a tree iff it has a unique path between every two vertices.

Problem 3.
Let G be a weighted graph and suppose there is a unique edge $e \in E(G)$ with smallest weight, that is, $w(e) < w(f)$ for all edges $f \in E(G) - \{e\}$. Prove that any minimum weight spanning tree (MST) of G must include e.

Problem 4.

A simple graph, G, is said to have *width* 1 iff there is a way to list all its vertices so that each vertex is adjacent to at most one vertex that appears earlier in the list. All the graphs mentioned below are assumed to be finite.

(a) Prove that every graph with width one is a forest.

Hint: By induction, removing the last vertex.

(b) Prove that every finite tree has width one. Conclude that a graph is a forest iff it has width one.