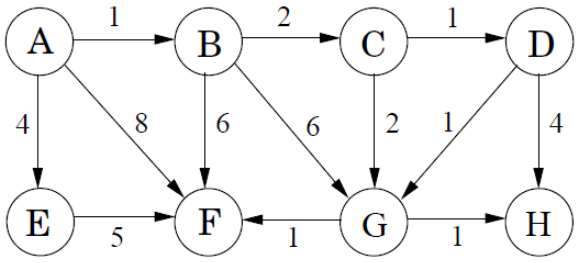
Homework 1

Due at 2pm, Feb 17, 2015.

Problem 1. Fill the following blanks using O, o, Ω, ω, or Θ. If both O and o are correct, use o; if both Ω and ω are correct, use ω; if both O and Ω are correct, use Θ. To distinguish O and o in handwriting, please write "Big-O" for O and "small-o" for o.

1. $n = \_\_\_\_(10n)$.
2. $0.1n = \_\_\_\_(10 \log n)$.
3. $2^n = \_\_\_\_(n^3)$.
4. $2n \log n = \_\_\_\_(n^2)$.

Problem 2. Suppose Dijkstra's algorithm is run on the following graph, starting at node A.



1. Draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm.
2. Show the final shortest-path tree.

Problem 3.

1. Someone claims the following property. Suppose that we are given an undirected and connected graph G, with weights $w(e)$ on all edges e. For any cycle C, if there is an edge e on this cycle with weight $w(e) > w(e')$ for all other edges $e' \in C$, then no MST contains e.

   Do you think the property is correct? If yes, prove it. Otherwise, give a counterexample.

2. The person further proposes the following algorithm for MST, assuming that the edge weights are all distinct. Starting from edge set E, the algorithm keeps taking a cycle C (if any) and removing an edge $e \in C$ with the largest weight. The algorithms stops when no cycle is found, at which point it outputs the remaining edges.

   Do you think the algorithm is correct? If yes, prove it. Otherwise, give a counterexample.

Problem 4*. (*ELITE students need to do this. Regular students are welcome to do it for bonus points.*)

An undirected graph $G = (V, E)$ is 3-colorable if there is a coloring $c : V \to \{1, 2, 3\}$ s.t. any two adjacent vertices have different colors. Suppose that a given graph G is 3-colorable.

1. Show that there also exists a 2-coloring $c : V \to \{1, 2\}$ s.t. no triangle in the graph have the three vertices sharing the same color.
2. Consider the following algorithm for finding a 2-coloring. The algorithm begins with an arbitrary 2-coloring c and repeats the following. While there is a triangle with the three vertices sharing the same color, randomly pick one of the three vertices and flip its color (from 1 to 2, or from 2 to 1). Do you think this algorithm stops in polynomial time in expectation?