Algorithms Final Examination
Jan. 15, 2014

1. (10% Ch. 16) What is an optimal Huffman code for the following set of frequencies, based on the first 6 Fibonacci numbers? a:1 b:1 c:2 d:3 e:5 f:8
Can you generalize your answer to find the optimal code when the frequencies are the first n Fibonacci numbers?

2. (10% Ch. 17) What is the total cost of executing n of the stack operations PUSH, POP, and MULTIPOP, assuming that the stack begins with s objects and finishes with f objects?

3. (10% Ch. 22) Give a linear-time algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two vertices s and t, and returns the number of simple paths from s to t in G. (Your algorithm needs only to count the number of simple paths, not list them.)

4. (10% Ch. 23) Suppose that all edge weights in a graph are integers in the range from 1 to $|V|$. How fast can you make Kruskal's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W?

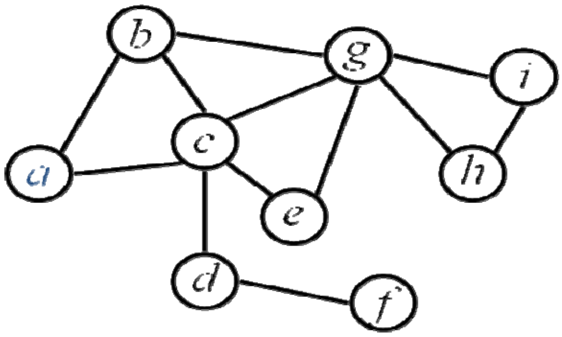
5. (10% Ch. 24) Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers to solve the single-source shortest paths problem.

6. (10% Ch. 24) Suppose that we are given a weighted, directed graph $G=(V, E)$ in which edges that leave the source vertex s may have negative weights, all other edge weights are nonnegative, and there are no negative-weight cycles. Argue that Dijkstra's algorithm correctly finds shortest paths from s in this graph.

7. (10% Ch. 25) Suppose that we run Johnson's algorithm on a directed graph G with weight function w. Show that if G contains a 0-weight cycle c, then $\hat{w}(u, v) = 0$ for every edge (u, v) in c.

8. (10% Ch. 25) Given an $O(VE)$-time algorithm for computing the transitive closure of a directed graph $G=(V, E)$.

9. (10% Ch. 35) Please give an approximation algorithm to select the minimum number of vertices in the following graph such that each edge has at least one vertex selected. Show the optimal solution is at most two times of your solution.



10. (10% Ch. 34) For each of the following statements, determine whether it is true or false. If the statement is correct, briefly state why. If the statement is wrong, explains why. Your answer will be evaluated based on your explanation and not the True/False marking alone.

    a. If a problem *A* is polynomial time reducible to problem *B* and *B* has a polynomial time algorithm, then problem *A* has a polynomial time algorithm.
    b. If we can prove that problem *A* can polynomial-time reduce to satisfiability problem, then problem *A* is NP-complete.
    c. Any NP-complete problem cannot be solved in polynomial time with non-deterministic Turing machine.
    d. Any NP-complete problem can be solved in polynomial time if there is an algorithm that can solve the 2-CNF satisfiability problem in polynomial time.
    e. Any NP-hard problem can be solved in polynomial time if there is an algorithm that can solve the satisfiability problem in polynomial time.