George Washington University
Department of Computer Science

CS212 - Midterm Exam

2 hours and 30 minutes (closed book)
10/23/2013

Total 100 points.

1. (5 pts. each) Let $n \geq 1$.

(a) Using the definitions of O (big-Oh) and Ω (Omega) notations, prove that:

$$\text{If } g(n) = \Omega((f(n)) \text{ and } g(n) = O(h(n)), \text{ then } f(n) = O(h(n)).$$

(b) Let $T(n) = cn + T(\frac{n}{2})$ for $n > 1$ and $T(1) = c$ where c is a constant. Show that $T(n) = O(n)$. Assume $n = 2^k$ for some integer k.

(c) Let $T(n) = cn + 2T(\frac{n}{2})$ for $n > 1$ and $T(1) = c$ where c is a constant. Show that $T(n) = O(n \log n)$. Assume $n = 2^k$ for some integer k.

2. (5 pts) Construct a set of Hoffman codes for an alphabet with 7 characters $a_1, \cdots, a_7$ with relative frequencies $(q_1, \cdots, q_7) = (3, 3, 7, 7, 8, 15, 19)$.

3. (5 pts. each) Suppose you are given an array of n element in increasing order.

(a) Describe an $O(n)$ time algorithm to find two numbers from the list that add up to zero, or report there is no such pair.

(b) Describe an $O(n^2)$ time algorithm to find three numbers from the list that add up to zero, or report that there is no such triple.

4. (10 pts) Consider the fractional Knapsack problem discussed in class defined as: given a knapsack with capacity M and n items with weight function $w(i)$ and profit function $p(i)$ for each i, $1 \leq i \leq n$, the objective is to find a solution $X = (x_1, \cdots, x_n)$ such that (i) $0 \leq x_i \leq 1$, (ii) $\sum_{i=1}^{n} x_i w(i) \leq M$, and (iii) $\sum_{i=1}^{n} x_i p(i)$ is maximized.

   Now, we consider a slight modification of the problem such that we are given an upper $u(i)$ for each $x(i)$, $1 \leq i \leq n$, i.e., $0 \leq x(i) \leq u(i)$ for each i. Note that the original Knapsack problem is when $u(i) = 1$ for each i.

   Design a greedy algorithm to solve this problem, and justify the correctness of your algorithm. (In your argument, you may assume the correctness of the greedy algorithm solving the original version of the Knapsack problem.)

5. (5 pts each) (Show all your work.)

(a) Apply the Partition algorithm to $A = (3, 3, 8, 5, 3, 8, 10, 3, 2, 3, 3, 1, 4, 3)$ using the first element as the pivot. In your output, elements with the same value of the pivot element **must** be placed in the left of the pivot element upon completion of the Partition algorithm.

(b) Apply the linear-time Select algorithm to the following example:
$L = \{1, 3, 13, 14, 2, 6, 18, 5, 15, 16, 17, 18, 12, 7, 8, 18, 12, 18, 19, 20, 21, 22\}$ and $k = 18$.

6. Consider a sequence $x_1, x_2, \cdots, x_n$ of numbers such that for some unknown i ($1 \leq i \leq n$, $x_1 < x_2 < \cdots < x_i$ and $x_i > x_{i+1} > \cdots > x_n$.

(a) (5 pts) Describe an $O(\log n)$ time algorithm to find the kth smallest element for an arbitrary integer k.

(b) (5 pts) Apply your algorithm in (a) to the following example: $X = (3, 5, 6, 8, 10, 9, 7, 4, 2, 1)$ and $k = 6$. (Show all your works.)

7. (5 pts.) Given n segments of line (into the X axis) with coordinates $[l_i, r_i]$, i.e., left and right end points. You are to choose the minimum number of line segments that cover the segment $[0, M]$. Design a greedy algorithm to solve this problem and argue why your algorithm is optimal.

8. (5 pts) A *forest* is defined to be a graph without a cycle, i.e., a graph that may have more than one component, but each component does not include a cycle. Let $G(V, E)$ be a connected edge-weighted graph. Let k be an arbitrary integer in $1 \leq k \leq |V|$.

(a) Design an algorithm for finding a forest with exactly k components such that the sum of the weights of edges in the forest is minimum.

(b) Let $V_0 = \{v_1, v_2, \cdots, v_k\}$ be a subset of k vertices in V. Design an algorithm to find a forest with exactly k components such that each component contains exactly one of the vertices in V_0.

9. (10 pts) Assume that edge weights are distinct in a given graph. Prove that there exists an unique minimum spanning tree of G.

10. (5 pts each) Consider the following graph G. Show all your work for each of the following problems.

(a) Find a minimum spanning tree of G obtained using the Prim's algorithm starting from node 3.

(b) Find a minimum spanning tree of G obtained using the Kruskal's algorithm.

(c) Apply the Dikstra's shortest path algorithm and find a shortest path tree of G where node 3 is the start node.

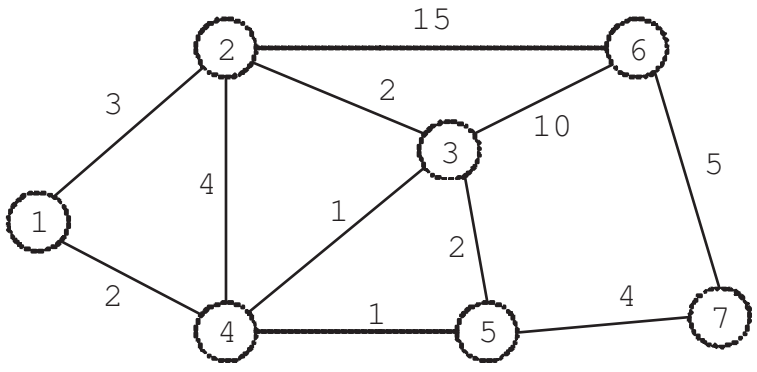

Figure 1: G