EECS 336 In-Class Test #1

02/12/2015

Problem 1 (20 points).

(a) **(6 points)** Use the master theorem to solve the following recurrences. You do not need to prove the correctness of your answers.

Recurrence	**Asymptotic Tight Bound**
$T(n) = 4T(n/2) + 5n^2$	$T(n) = \Theta(\underline{\hspace{2cm}})$
$T(n) = 3T(n/2) + n \log n$	$T(n) = \Theta(\underline{\hspace{2cm}})$
$T(n) = 8T(n/3) + n^2 \log^3 n$	$T(n) = \Theta(\underline{\hspace{2cm}})$

(b) **(6 points)** For every integer $n \geq 1$, let $f(n)$ be the maximum integer k such that $k^k \leq n$. Prove that $f(n) = O(\lg n / \lg \lg n)$.

(c) **(8 points)** Guess the asymptotic tight bound for the recurrence $T(n) = 3T(n-1) + 3^n$. Then use the substitution method to prove that your guess is an asymptotic upper bound.

Problem 2 (10 points). Given an array A of n numbers, a number x is a *weak majority* of A if it appears in A at least $\lceil n/3 \rceil$ times. For example, if $A = (30, 18, 12, 32, 19, 54, 18, 32, 18)$ is an array of length $n = 9$, then the number 18 is a weak majority of A since it appears in A 3 times. The number 32 is not a weak majority since it appears in A only twice. There might be $0, 1, 2$ or 3 distinct weak majorities of a given array A.

Give a deterministic $O(n)$-time algorithm that checks if A contains a weak majority. You can use any algorithms you learnt in class as sub-routines.

Problem 3 (15 points). Consider the following algorithm for finding the minimum spanning tree of a connected graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$.

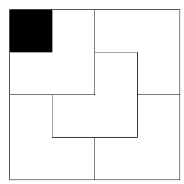
(1) $T \leftarrow E$
(2) **while** T is not a tree
(3)   let C be an arbitrary cycle in T
(4)   let e be the heaviest edge in C (ties are broken arbitrarily)
(5) $T \leftarrow T \setminus \{e\}$

(6) **return** T

Does the algorithm always return a minimum spanning tree T of G? If your answer is yes, prove it; if your answer is no, give an instance (G, w) for which the algorithm does not return a minimum spanning tree.

Problem 4 (15 points). *To get the full 15 points, you only need to solve* ***either (a) or (b)****. If you write down solutions for both (a) and (b), then your score for this problem will be the* ***maximum*** *of your score for (a) and your score for (b).* ***Use your time wisely.***

(a) **(15 points)** Consider a $2^n \times 2^n$ chessboard with one arbitrarily chosen square removed. Prove that the chessboard can be tiled without gaps by L-shaped pieces, each composed of 3 squares. An L-shaped piece can be rotated; thus there are 4 different orientations of an L-shaped piece. For example, the following figure shows a tiling of a 4×4 chessboard with the top-left square removed, using 5 L-shaped pieces.



(b) **(15 points)** Let $G = (V, E)$ be a directed acyclic graph with $V = \{1, 2, 3, \cdots, n\}$. Suppose all directed edges $(i, j) \in E$ have $i < j$. Let $w : E \to \mathbb{R}$ be a weight function. Give an $O(E)$-time algorithm that computes the length of the second shortest path from 1 to n in G. If there are two different shortest paths, then the length of the second shortest path is the same as the length of the shortest path.

Problem 5 (40 points). Given an array A of n numbers, we say that a 10-tuple $(i_1, i_2, \cdots, i_{10})$ of integers is inverted if $1 \le i_1 < i_2 < i_3 < \cdots < i_{10} \le n$ and $A[i_1] > A[i_2] > A[i_3] > \cdots > A[i_{10}]$.

(a) **(15 points)** Give an $O(n^2)$-time algorithm to count the number of inverted 10-tuples w.r.t A.

(b) **(25 points)** Give an $O(n \lg n)$-time algorithm to count the number of inverted 10-tuples w.r.t A.