15-451 Algorithms, Spring 1997
Midterm Exam

February 27, 1997

Instructions: This exam is open book and open notes. Don't share anything with your neighbor. There are problems on both sides of this page. Write your name on anything you turn in.

Problem 1 (42 points = 6×7)

For each of the following statements, state whether it is true or false and give an explanation of at most 5 lines.

a) Checking whether an array $A[1..n]$ is in sorted order (i.e., whether $A[1] \leq A[2] \leq A[3] \leq \ldots \leq A[n]$) can be done in $O(n)$ time.

b) **(i)** The function $n \log_2 n$ is $O(n \log_{10} n)$.
(ii) The function $n^{\log_2 n}$ is $O(n^{\log_{10} n})$.

c) In doing Union-Find with path compression and balancing, the cost of *each individual* FIND instruction is $O(\log^* n)$ where n is the number of keys.

d) Merging n sorted lists, each containing 10 elements, into 1 sorted list using only comparisons takes $\Omega(n \log n)$ comparisons.

e) If an array $A[1..n]$ of integers is sorted, then for any two integers x, y, we can determine the number of keys (among $A[1..n]$) in the interval $[x,\ y]$ in $O(\log n)$ time.

f) If $A[1..n]$ stores a heap (i.e., we have $A[i] \geq A[2i]$ and $A[i] \geq A[2i+1]$ whenever in range), then the second largest element can be found in $O(1)$ time.

Problem 2 (14 points)

Given a binary search tree, T, devise an algorithm to determine whether the tree is an **AVL tree**. Your algorithm should run in $O(n)$ time, where n is the number of nodes in the tree. Argue that any algorithm that solves this problem has running time $\Omega(n)$.

Problem 3 (20 points)

Suppose we modify the deterministic linear time selection algorithm to divide the n elements into $n/7$ sets of 7 elements each, (assume n is a power of 7), sort each 7 element set and pivot about the median of the "middle" $n/7$ elements. Argue that the running time $T(n)$ satisfies the recurrence:

$$T(n) \leq T(n/7) + T(5n/7) + cn$$

Assuming the above recurrence (even if you were not able to argue that it holds), show by induction on n that we have

$$T(n) \leq 7cn$$

Problem 4 (24 points)

You are to drive a car for n miles. The capacity of the gas tank in the car is b (gallons). There are $n/30$ gas stations on the way, at the 0^{th}, 30^{th}, 60^{th}, 90^{th}, etc. miles. At each gas station, you may add an integral number of gallons of gas to your tank without exceeding the capacity. The car uses one gallon of gas each 30 miles. You are given the price per gallon of gas at each gas station. Give an $O(nb^2)$ time algorithm to find the least cost way of driving the n miles starting at the first gas station with an empty tank. Argue convincingly that the program is correct and runs in the desired time.

Note: If you are stuck on this problem, you can request a short, standard hint from the TA at the front of the room. Receiving the hint will deduct 7 points from your score *on this problem.* You won't be given less than 0 even if you take the hint.