CSE 101 - ALGORITHMS - SUMMER 2000
Quiz 1

Wednesday, July 12th.

Problem 1: (2 points)

The following algorithm to test whether a number q is prime was presented in the first lecture:

PRIME(q)
1 **for** $i \leftarrow 2$ **to** $\sqrt{q}$ **do**
2 **if** i $divides$ q **then return** NO
3 **return** YES

Show that PRIME is correct, that is, that $\sqrt{q}$ is a correct upper bound for potential divisors of q.

Proof: It suffices to show that if q has divisors different than 1 and q then it has at least one divisor between 2 and $\sqrt{q}$. We prove it by contradiction: suppose that $q = x \cdot y$ and that $x > \sqrt{q}$ and $y > \sqrt{q}$; then $x \cdot y > \sqrt{q} \cdot \sqrt{q} = q$, that is, $q > q$, contradiction.

Problem 2: (2 points)

A divide an conquer algorithm $\mathcal{A}$ splits the original problem in two related subproblems of size $\sqrt{n}$ and then needs $\log n$ time to combine the solutions of the subproblems into a solution for the original problem. What's the running time of $\mathcal{A}$?

Proof: The recurence for the running time of $\mathcal{A}$ is (1 point)

$$T(n) = 2T(\sqrt{n}) + \log n.$$

Let's solve this recurence (1 point) now. Substituting $n = 2^m$, we get that $T(2^m) = 2T(2^{\frac{m}{2}}) + m$; notice that $m = \log n$. Further, substituting $S(m) = T(2^m)$, we obtain the new recurence $S(m) = 2S(m/2) + m$ which is the same as the one for merge sort, so its solution is $S(m) = m \log m$. Substituting back $S(m) = T(2^m)$ and $m = \log n$, we obtain that $T(n) = \log n \cdot \log \log n$.

Problem 3: (6 points)

Let $A = \langle 3, 9, 5, 3, 1, 4, 8, 7 \rangle$. Illustrate the execution of

1. INSERTION-SORT(A);
2. SELECTION-SORT(A);
3. QUICK-SORT($A, 1, 8$); do *not* illustrate the execution of PARTITION;
4. MERGE-SORT($A, 1, 8$); do *not* illustrate the execution of MERGE;
5. HEAP-SORT(A); do *not* illustrate the executions of BUILD-HEAP and HEAPIFY;
6. COUNTING-SORT($A, 9$).

Proof: We only show how the array A is modified. This is enough to show that you understood the sorting algorithms. There were many different solutions in your quizes; I condidered them all correct if it was clear that you understood the algorithms.

1. INSERTION-SORT
   $\langle 3, \underline{9}, 5, 3, 1, 4, 8, 7\rangle$
   $\langle 3, 9, \underline{5}, 3, 1, 4, 8, 7\rangle$
   $\langle 3, 5, 9, \underline{3}, 1, 4, 8, 7\rangle$
   $\langle 3, 3, 5, 9, \underline{1}, 4, 8, 7\rangle$
   $\langle 1, 3, 3, 5, 9, \underline{4}, 8, 7\rangle$
   $\langle 1, 3, 3, 4, 5, 9, \underline{8}, 7\rangle$
   $\langle 1, 3, 3, 4, 5, 8, 9, \underline{7}\rangle$
   $\langle 1, 3, 3, 4, 5, 7, 8, 9\rangle$

2. SELECTION-SORT
   $\langle 3, 9, 5, 3, 1, 4, 8, 7\rangle$
   $\langle \underline{1}, 9, 5, 3, 3, 4, 8, 7\rangle$
   $\langle 1, \underline{5}, 9, 3, 3, 4, 8, 7\rangle$
   $\langle 1, \underline{3}, 9, 5, 3, 4, 8, 7\rangle$
   $\langle 1, 3, \underline{5}, 9, 3, 4, 8, 7\rangle$
   $\langle 1, 3, \underline{3}, 9, 5, 4, 8, 7\rangle$
   $\langle 1, 3, 3, \underline{5}, 9, 4, 8, 7\rangle$
   $\langle 1, 3, 3, \underline{4}, 9, 5, 8, 7\rangle$
   $\langle 1, 3, 3, 4, \underline{5}, 9, 8, 7\rangle$
   $\langle 1, 3, 3, 4, 5, \underline{8}, 9, 7\rangle$
   $\langle 1, 3, 3, 4, 5, \underline{7}, 9, 8\rangle$
   $\langle 1, 3, 3, 4, 5, 7, \underline{8}, 9\rangle$

3. QUICK-SORT

   The array A is first partitioned as (consider that the pivot is $A[1]$) $\langle 1, 3, \underline{3}, 9, 5, 4, 8, 7\rangle$. Then QUICK-SORT$(A, 1, 3)$ and QUICK-SORT$(A, 4, 8)$ are called. The first is not going to change the array A, so we only illustrate the second. The subarray $\langle \underline{9}, 5, 4, 8, 7\rangle$ is again partitioned (the pivot is 9 now) modifying A to $\langle 1, 3, 3, 7, 5, 4, 8, 9\rangle$ and then QUICK-SORT$(A, 4, 7)$ is called. We keep doing this and obtain $\langle 1, 3, 3, 4, 5, \underline{7}, 8, 9\rangle$.

4. MERGE-SORT
   $\langle 3, 9, 5, 3, 1, 4, 8, 7\rangle$
   $\langle 3, 9, 5, 3\rangle$ $\langle 1, 4, 8, 7\rangle$
   $\langle 3, 9\rangle$ $\langle 5, 3\rangle$ $\langle 1, 4\rangle$ $\langle 8, 7\rangle$
   $\langle 3\rangle$ $\langle 9\rangle$ $\langle 5\rangle$ $\langle 3\rangle$ $\langle 1\rangle$ $\langle 4\rangle$ $\langle 8\rangle$ $\langle 7\rangle$
   $\langle 3, 9\rangle$ $\langle 3, 5\rangle$ $\langle 1, 4\rangle$ $\langle 7, 8\rangle$
   $\langle 3, 3, 5, 9\rangle$ $\langle 1, 4, 7, 8\rangle$
   $\langle 1, 3, 3, 4, 5, 7, 8, 9\rangle$

5. HEAP-SORT

   The procedure BUILD-HEAP yields $A = \langle 9, 7, 8, 3, 1, 4, 5, 3\rangle$. Then the following changes generated by swapings and heapifies end up with the sorted array:

   $\langle 3, 7, 8, 3, 1, 4, 5, \underline{9}\rangle$
   $\langle 8, 7, 5, 3, 1, 4, 3, \underline{9}\rangle$
   $\langle 3, 7, 5, 3, 1, 4, \underline{8}, \underline{9}\rangle$
   $\langle 7, 3, 5, 3, 1, 4, \underline{8}, \underline{9}\rangle$
   $\langle 4, 3, 5, 3, 1, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 5, 3, 4, 3, 1, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 1, 3, 4, 3, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 4, 3, 1, 3, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 3, 3, 1, \underline{4}, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 1, 3, \underline{3}, \underline{4}, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle 1, \underline{3}, \underline{3}, \underline{4}, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$
   $\langle \underline{1}, \underline{3}, \underline{3}, \underline{4}, \underline{5}, \underline{7}, \underline{8}, \underline{9}\rangle$

6. COUNTING-SORT

   The frequency array is $F = \langle 1, 0, 2, 1, 1, 0, 1, 1, 1\rangle$. After the next step, it becomes $F = \langle 1, 1, 3, 4, 5, 5, 6, 7, 8\rangle$. Next, we visit the elements in A from the last one toward the first and output them in B according to F, appropriately decreasing the frequencies. We get $B = \langle 1, 3, 3, 4, 5, 7, 8, 9\rangle$ and $F = \langle 0, 1, 1, 3, 4, 5, 5, 6, 7\rangle$.