Practice Quiz 1

- Do not open this quiz booklet until you are directed to do so. Read all the instructions first.
- When the quiz begins, write your name on every page of this quiz booklet.
- The quiz contains 5 multi-part problems. You have 80 minutes to earn 80 points.
- This quiz booklet contains 10 pages, including this one. Two extra sheets of scratch paper are attached. Please detach them before turning in your quiz.
- This quiz is closed book. You may use one handwritten A4 or $8\frac{1}{2}'' \times 11''$ crib sheet. No calculators or programmable devices are permitted.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Do not put part of the answer to one problem on the back of the sheet for another problem, since the pages may be separated for grading.
- Do not waste time and paper rederiving facts that we have studied. It is sufficient to cite known results.
- Do not spend too much time on any one problem. Read them all through first, and attack them in the order that allows you to make the most progress.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Good luck!

Problem	Points	Grade	Initials
1	12		
2	12		
3	30		
4	13		
5	13		
Total	80		

Name: ______________________________

Problem 1. Recurrences [12 points]

Solve the following recurrences. Give tight, i.e. $\Theta(\cdot)$, bounds.

(a) $T_1(n) = 5\,T_1(n/2) + \sqrt{n}$

(b) $T_2(n) = 64\,T_2(n/4) + 8^{\lg n}$

(c) Set up the recurrence for Strassen's matric multiplication algorithm and solve it.

Problem 2. Short Answer [12 points]

Give *brief*, but complete, answers to the following questions.

(a) Briefly describe the difference between a ***deterministic*** and a ***randomized*** algorithm, and name two examples of algorithms that are not deterministic.

(b) Describe the difference between ***average-case*** and ***worst-case*** analysis of deterministic algorithms, and give an example of a deterministic algorithm whose average-case running time is different from its worst-case running time.

(c) If you can multiply 4-by-4 matrices using 48 scalar multiplications, can you multiply $n \times n$ matrices asymptotically faster than Strassen's algorithm (which runs in $O(n^{\lg 7})$ time)? Explain your answer.

Problem 3. True or False, and Justify [30 points]

Circle **T** or **F** for each of the following statements, and briefly explain why. The better your argument, the higher your grade, but be brief. Your justification is worth more points than your true-or-false designation.

(a) T F Every comparison-based sort uses at most $O(n \log n)$ comparisons in the worst case.

(b) T F RADIX-SORT is stable if its auxiliary sorting routine is stable.

(c) T F It is possible to compute the smallest $\sqrt{n}$ elements of an n-element array, in sorted order, in $O(n)$ time.

(d) T F Consider hashing the universe $U = \{0, \ldots, 2^r - 1\}$, $r > 2$, into the hash table $\{0, 1\}$. Consider the family of hash functions $\mathcal{H} = \{h_1, \ldots, h_r\}$, where $h_i(x)$ is the ith bit of the binary representation of x. Then $\mathcal{H}$ is universal.

(e) T F RANDOMIZED-SELECT can be forced to run in $\Omega(n \log n)$ time by choosing a bad input array.

(f) T F For every two functions $f(n)$ and $g(n)$, either $f(n) = O(g(n))$ or $g(n) = O(f(n))$.

(g) **T F** Let H be a universal hash family mapping keys into a table of size $m = n^2$. Then, if we use random $h \in H$ to hash n keys into the table, the expected number of collisions is at most $1/n$.

(h) **T F** The following array A is a max-heap:

$$30 \;\; 25 \;\; 7 \;\; 18 \;\; 24 \;\; 8 \;\; 4 \;\; 9 \;\; 12 \;\; 22 \;\; 5$$

(i) **T F** Suppose we use HEAPSORT instead of INSERTION-SORT as a subroutine of BUCKET-SORT to sort n elements. Then BUCKET-SORT still runs in average-case linear time, but its worst-case running time is now $O(n \log n)$.

(j) **T F** If memory is limited, one would prefer to sort using HEAPSORT instead of MERGESORT.

Problem 4. Mode finding [16 points]

Assume that you are given an array $A[1 \dots n]$ of distinct numbers. You are told that the sequence of numbers in the array is *unimodal*, i.e., there is an index i such that the sequence $A[1 \dots i]$ is increasing (i.e. $A[j] < A[j+1]$ for $1 \leq j < i - 1$) and the sequence $A[i \dots n]$ is decreasing. The index i is called the *mode* of A.

Show that the mode of A can be found in $O(\log n)$ time.

Problem 5. Assigning Grades [13 points]

It is the not-too-distant-future, and you are a computer science professor at a prestigious north-eastern technical institute. After teaching your course, "6.66: Algorithms from Hell," you have to assign a letter grade to each student based on his or her unique total score. (Scores can only be compared to each other.) You are grading on a curve, and there are a total of k different grades possible. You want to rearrange the students into k equal-sized groups, such that everybody in the top group has a higher score than everybody in the second group, etc. However, you don't care how the students are ordered within each group (because they will all receive the same grade).

(a) Describe and analyze a simple algorithm that takes an unsorted n-element array A of scores and an integer k, and divides A into k equal-sized groups, as described above. Your algorithm should run in time $O(nk)$. (If you find a faster algorithm, see part **(c)**.) You may assume that n is divisible by k. *Note*: k is an input to the algorithm, not a fixed constant.

(b) In the case that $k = n$, prove that any algorithm to solve this problem must run in time $\Omega(n \log k)$ in the worst case. Recall that we are only considering comparison-based algorithms, i.e., algorithms that only compare scores to each other as a way of finding information about the input. *Hint:* There is a very short proof.

(c) Now describe and analyze an algorithm for this problem that runs in time $O(n \log k)$. You may also assume that k is a power of 2, in addition to assuming that n is divisible by k.

SCRATCH PAPER — Please detach this page before handing in your quiz.

SCRATCH PAPER — Please detach this page before handing in your quiz.