

Name: _____

ID No: _____

CSE 361 Complexity of Sequential and Parallel Algorithms

Fall 2005 Exam I

Note: You are supposed to give proofs to the time bounds of your algorithms. Read the questions carefully before attempting to solve them.

1. (17 points) Input is an array A of size n . The array can only be of one of the following two types:

- **Type I:** A has $\frac{3}{4}n$ ones and $\frac{1}{4}n$ zeros.
- **Type II:** A has $\frac{1}{4}n$ ones and $\frac{3}{4}n$ zeros.

How will you identify the type of A in $O(\log n)$ time using a Monte Carlo algorithm? Show that the probability of a correct answer will be $\geq 1 - n^{-\alpha}$ for any constant $\alpha \geq 1$.

- (16 points) Input are an array $a[]$ of n elements and an element x . We have to check if there are three elements in $a[]$ whose sum is x . Assume that the array elements are distinct. Present an $O(n^2 \log n)$ time algorithm for this problem.

3. A_1, A_2, \dots, A_m are sorted sets such that $\sum_{i=1}^m |A_i| = n$.

(a) (9 points) Present an $O(n)$ time algorithm to compute $A_1 \cap A_2 \cap \dots \cap A_m$.

(b) (8 points) Present an $O(n \log m)$ time algorithm to compute $A_1 \cup A_2 \cup \dots \cup A_m$.

4. (17 points) Input is a sequence X of n numbers, not necessarily in sorted order. The problem is to find an element $k \in X$ such that

$$\sum_{x \in X \text{ and } x \leq k} x \geq \frac{1}{4} \sum_{y \in X} y,$$

i.e., the sum of all the elements of X that are $\leq k$ is at least one-fourth of the sum of all the elements of X . Also,

$$\sum_{x \in X \text{ and } x \geq k} x \geq \frac{3}{4} \sum_{y \in X} y.$$

For example, if the input is 5, 11, 8, 7, 27, 2 then the answer is 8. Present an $O(n)$ time algorithm to identify k .

5. (16 points) Input are two matrices A and B . A is of size $kn \times n$ and B is of size $n \times kn$. Show how to compute AB in time $O(k^2 n^{\log 7})$.

6. (17 points) $A[1 : n]$ is an array of distinct real numbers such that the difference between any two numbers in A is at least $\frac{1}{n^5}$ and at most n^5 . Present an $O(n)$ time algorithm to sort A .