CSE 5500 Algorithms. Fall 2018

Home work 1. Due on October 10, 2018.

1. There are $\sqrt{n}$ copies of an element in the array c. Every other element of c occurs exactly once. If the algorithm RepeatedElement is used to identify the repeated element of c, will the run time still be $\tilde{O}(\log n)$? If so, why? If not, what is the new run time?

2. Let $\mathcal{A}$ be a Monte Carlo algorithm that solves a decision problem π in time T. The output of $\mathcal{A}$ is correct with probability c, c being a constant greater than 1/2. Show how you can modify $\mathcal{A}$ so that its answer is correct with high probability. The modified version can take $O(T \log n)$ time.

3. Input is a sequence X of n keys with many duplications such that the number of distinct keys is d $(< n)$. Present an $O(n \log d)$-time sorting algorithm for this input. (For example, if $X = 5, 6, 1, 18,\ 6, 4, 4, 1,\ 5, 17$, the number of distinct keys in X is six.)

4. Input is an array of n arbitrary real numbers (where n is odd). The array has $(n+1)/2$ distinct numbers such that each number has exactly two copies excepting for one number. Present an $O(n)$ time algorithm to identify the unique number.

5. Input is a (not necessarily sorted) sequence $S = k_1, k_2, \ldots, k_n$ of n arbitrary numbers. Consider the collection C of n^2 numbers of the form $\min\{k_i, k_j\}$, for $1 \leq i, j \leq n$. Present an $O(n)$-time and $O(n)$-space algorithm to find the median of C.

6. Two sets A and B have n elements each. Assume that each element is an integer in the range $[0, n^{100}]$. These sets are not necessarily sorted. Show how to check whether these two sets are disjoint in $O(n)$ time. Your algorithm should use $O(n)$ space.

7. Two rooted trees T_1 and T_2 are said to be *isomorphic* if there exists a one-to-one mapping f from the vertices of T_1 to those of T_2 satisfying the following condition: for each internal vertex v of T_1 with the children $v_1, \ldots, v_k$, the vertex $f(v)$ has as children exactly the vertices $f(v_1), f(v_2), \ldots, f(v_k)$. Observe that no ordering is assumed on the children of any internal vertex. Devise an efficient randomized algorithm for testing the isomorphism of rooted trees and analyze the performance.

8. Given a randomized algorithm for testing the existence of a perfect matching in a graph G, describe how you would actually construct such a matching. What is the run time of your algorithm if you use the testing algorithm described in class?

9. Consider the two-dimensional version of the pattern matching problem. The text is an $n \times n$ Boolean matrix X and the pattern is an $m \times m$ Boolean matrix Y. A pattern match occurs if Y appears as a (contiguous) sub-matrix of X. To apply the randomized algorithm described in class, we can convert Y into an m^2-bit vector using the row-major format. The possible occurrences of Y in X are the m^2-bit vectors $X(j)$ obtained by taking all $(n-m+1)^2$ sub-matrices of X in a row-major form. It is clear that the algorithm discussed in class can be used in this case. Analyze the error probability and run time in this case.