CS21003 - Tutorial 8

October 20th, 2017

1. Let $t \in N$ be a constant and b a constant base (like 10 or 16). You are given an array of n integers, each in the range 0 to $b^t - 1$. Modify the radix sort algorithm to sort the input array in $O(n)$ time.
2. You are given an array of n dates in the $dd - mm - yyyy$ format. Propose a linear-time algorithm to sort the array in the usual increasing order (chronological order).
3. Suppose that n points are chosen uniformly inside a circle of radius r (that is, the probability of choosing a point in any region R of area a inside the circle is $a/(\pi r^2)$). Give an algorithm that sorts the n given points with respect to their distances from the center of the circle in expected linear time.
4. Let $A = (a_0, a_1, a_2, \ldots, a_{n-1})$ be an unsorted array of n floating-point numbers. Propose an $O(n)$-time algorithm to compute the (floating-point) number x (not necessarily an element of A) for which $max_{0 \leq i \leq n-1} |a_i - x|$ is as small as possible. Prove the correctness of your algorithm.
5. You are given two alphabetic (lower case) strings S and T each of the same length n. Propose an $O(n)$-time algorithm to decide whether S can be obtained by permuting the symbols of T. (Examples: The string *algorithm* is a permutation of *logarithm*, *retinae* is a permutation of *trainee* but not of *entrain* or *trainer*.)
6. Given n integers in the range 0 to k, describe an algorithm that pre-processes its input in $O(n + k)$ time, then answers an query about how many of the n integers fall into a range $[a : b]$ in $O(1)$ time. You can use $O(k)$ additional space.
7. Suppose we are working with proteins, that is $\Sigma = \{A, C, G, T\}$. We want to find all matches of TAG in CTAGTCTAGA. Show the iterations of Rabin-Karp algorithm to find all the matches. You can use the prime number for the hash to be 13.
8. Suppose you are given two string S and T, each of length n, and it is given that they share a common substring of length m. Show how you can solve this problem of finding that common substring in expected $O(n)$ time. [**Hint:** Use rolling hashes.]