Algorithms I

Tutorial 2

August 19, 2016

Problem 1

You are given an array a containing n distinct integers. You are also given multiple integers x_i. For each x_i, you need to find number of indices i such that $a[i] < x$. You are allowed to do pre-computation of $O(n \log n)$, but for each i, you should be able to answer in $O(\log n)$. You are not allowed to use sorting.

Problem 2

Suppose that $\{(a_1, b_1), (a_2, b_2) \ldots (a_n, b_n)\}$ is a set of n pairs of integers. Assume that all a_i and b_i values are distinct. Formally, for any i, j such that $1 \leq i, j \leq n$ and $i \neq j$, all a_i, b_i, a_j and b_j are distinct. You need to create a data structure to support search, insert and deletion in $O(\log n)$ time. Each search or deletion can be with respect the first component (a_i) or second component (b_i). i.e. Given an integer x, you need to find whether there exists a pair (a_i, b_i) with $a_i = x$ or $b_i = x$. For deletion, you delete the pair (a_i, b_i) with $a_i = x$ or $b_i = x$. Likewise for the second component.

Problem 3

Suppose that $\{(a_1, b_1), (a_2, b_2) \ldots (a_n, b_n)\}$ is a set of n pairs of integers. Assume that all a_i values are distinct and so are all the b_i values. For each (a_i, b_i), you need to find number of indices j such that $a_j < a_i$ and $b_j < b_i$. Your algorithm should run in $O(n \log n)$ overall.

Problem 4

You are given an array a of size n indexed as $1, 2, 3 \ldots n$. You are also given an integer k. You need to create array b such that $b[i] = max\{a[j] \mid max(i - k + 1, 1) \leq j \leq i\}$ in $O(n \log k)$.

Problem 5

Let $a_1, a_2 \ldots a_n$ be a sequence of integers with at most k distinct values. Design an $O(n \log k)$ algorithm to sort the sequence.

Problem 6

An array of size n with at most k distinct elements. Find m most frequent elements in $O(n \log k)$ using $O(k)$ additional space. Assume $m \leq k$. You are not allowed to use sorting. e.g. in array $[1, 2, 1, 1, 2, 3]$ if $m = 2$, the output will be $[1, 2]$

Problem 7

You are given k sorted arrays $A_1, A_2 \ldots A_k$. You need to merge these arrays to form another sorted array A. You should do it in $O(n \log k)$ where $n = \sum_{i=0}^{k} |A_i|$.

Problem 8

You are given with keys 10, 22, 31, 4, 15, 28, 17, 88, 59. You need to insert these keys into a hash table of length $m = 11$ using open addressing with the auxiliary hash function $h'(k) = k$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (kmod(m-1))$. Now, repeat the same procedure for $m = 22$.

Note: In the next two problems, you can assume that you have a hash table that supports search, insert and delete in expected $O(1)$

Problem 9

You are given an array a of n positive integers, and a target sum s. Your task is to find whether there exist distinct indices i, j such that $a[i] + a[j] = s$ in expected $O(n)$.

Problem 10

You are given an array a of n integers. Your task is to find the number of sub-arrays $a[i \ldots j]$ such that $\sum_{k=i}^{j} a[k] = 0$ in expected $O(n)$.