Design and Analysis of Algorithms : CS345/CS345A

Practice-sheet 1 :Divide and Conquer and Augmented Binary Search Trees

Date: *16th August, 2015*

1. Professor Math tells his class that it is asymptotically faster to square an n-bit integer than to multiply two n-bit integers. Should they believe him?

2. Given an array A storing n distinct numbers. A pair (i, j) where $0 \leq i < j \leq n-1$, is said to be an inversion if $A[i] > A[j]$. Design an $O(n \log n)$ time algorithm to count all inversions in A.

3. Suppose we want to evaluate the polynomial

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$$

at point x.

   (a) Show that the following simple routine, known as Horner's rule, does the job and leaves the answer in z.

   $z = a_n$
   for $i = n-1$ downto 0:
   $z = zx + a_i$

   (b) How many additions and multiplications does this routine use, as a function of n? Can you find a polynomial for which an alternative method is substantially better?

4. Let P be a set of n points in a plane. Design an $O(n \log n)$ algorithm to find the least perimeter triangle out of all possible triangles defined by P. Assume without loss of generality that there are no three collinear points in P.

5. Recall the problem of non-dominated points discussed in the class. The problem can be easily extended to three dimensions. Design an algorithm for this problem that takes $O(n \log n)$ time.
(The point $p_i(x_i,y_i,z_i)$ is dominated by $p_j(x_j,y_j,z_j)$ if $x_j > x_i$,$y_j > y_i$,$z_j > z_i$)

6. Given a list of values $z_0, \cdots, z_{n-1}$ (possibly with repetitions), show how to find the coefficients of the polynomial $P(x)$ of degree less than n that has zeros only at $z_0. \ldots, z_{n-1}$ (possibly with repetitions). Your procedure should run in time $O(n \log^2 n)$. (Hint: The polynomial $P(x)$ has a zero at z_j if and only if $P(x)$ is a multiple of $(x - z_j)$).

7. Show that any array of integers $x_1, x_2 \ldots x_n$ can be sorted in $O(n + M)$ time, where M is the difference between highest and lowest element of the array.
For small M, this is linear time, why doesn't the $\Omega(n \log n)$ lower bound apply in this case?

8. Given an array that represents elements of arithmetic progression in order. One element is missing in the progression, find the missing number in $O(\log n)$.

9. There are 2 sorted arrays A and B of size n each. Give a solution to find the median of the array obtained after merging the above 2 arrays(i.e. array of length 2n). The complexity should be $O(\log n)$

10. Consider two sets A and B, each having n integers in the range from 0 to $10n$. We wish to compute the **Cartesian** sum of A and B, defined by

$$C = \{x + y \ : \ x \in A \ \text{ and } \ y \in B\}.$$

Note that the integers in C are in the range from 0 to $20n$. We want to find the elements of C and the number of times each element of C is realized as a sum of elements in A and B. Design an $O(n \log n)$ time algorithm to achieve this objective.

11. A Toeplitz matrix is an $n \times n$ matrix $A = (a_{i,j})$ such that $a_{i,j} = a_{i-1,j-1}$ for $i = 2, 3, \ldots, n$ and $j = 2, 3, \ldots, n$.

    (a) Is the sum of two Toeplitz matrix necessarily Toeplitz ? What about the product ?

    (b) Describe how to represent a Toeplitz matrix so that two $n \times n$ matrices can be added in $O(n)$ time.

    (c) Give an $O(n \log n)$ time algorithm for multiplying an $n \times n$ Toeplitz matrix by a vector of length n. Use your representation from part (b).

    (d) Give an efficient algorithm for multiplying two $n \times n$ Toeplitz matrices. analyze its running time.

12. Maintain a data structure for storing a sequence of n numbers $\langle a_1, a_2, \cdots, a_n \rangle$ under the following operations.

    (a) Insert(i,b): Insert number b at ith position in the sequence.

    (b) Report(i): Return the number at ith position in the sequence.

    (c) Delete(i): Delete the number at ith position in the sequence.

    (d) Min(i,j) : report the smallest numbers among all the numbers that appear at positions from i to j in the sequence.

    (e) Add(i,j,x) : Add x to each numbers of the sequence from ith position to jth position.

    Each operation must take $O(\log n)$ worst case time.

13. The Josephus problem is defined as follows. Suppose that n people are arranged in a circle and that we are given a positive integer $m \leq n$. Beginning with a designated first person, we proceed around the circle, removing every mth person. After each person is removed, counting continues around the circle that remains. This process continues until all n people have been removed. The order in which the people are removed from the circle defines the (n, m)-Josephus permutation of the integers $1, 2, ..., n$. For example, the $(7, 3)$-Josephus permutation is $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$.

    (a) Suppose that m is a constant. Describe an $O(n)$-time algorithm that, given an integer n, outputs the (n, m)-Josephus permutation.

    (b) Suppose that m is not a constant. Describe an $O(n \log n)$ time algorithm that, given integers n and m, outputs the (n, m)-Josephus permutation.