Purdue University — Hambrusch

CS 381 — Spring 2015

Assignment 2

Due: Thursday, February 5, 2015 (hand in before class)

1) (8 pts.) Rank the running times shown below from smallest to largest and indicate if any are Θ of each other. Show your work and give precise explanations.

$\left(\frac{n}{2}\right)!$, $\left(\frac{9}{10}\right)^n$, $(3 \log\log n)^3$, $2^{n+\log n}$, $\left(\frac{\log n}{3}\right)^{1/3}$

2) (12 pts.)

(i) Use the Master Theorem to solve each of the following recurrence relations. Explain which case applies and why. $T(1) = 1$ holds for all. Assume n is either a power of 2, 4, or 5, as convenient.

1. $T(n) = 6T(n/2) + 5\sqrt{n}$
2. $T(n) = T(4n/5) + n \log n$
3. $T(n) = 16T(n/4) + n^3$
4. $T(n) = 3T(n/5) + 2n$

(ii) Use induction to show that $T(n) = 27T(n/3) + n^2 = O(n^3)$. Assume n is a power of 3 and $T(1) = 1$. Give precise lower bounds on constants used in the induction.

3) (12 pts.) Bonnie and Clyde each have n numbers. They know that the combined $2n$ numbers are unique. Performing any computation on their own numbers is cheap. Hence, the cost of sorting their own numbers or building a binary search tree is minimal and can be ignored. However, communication between Bonnie and Clyde is expensive. One unit of communication is sending one number; only the original numbers be communicated.

Design and describe an algorithm to find the n-th smallest number in the union of the $2n$ numbers which minimizes the total number of communications performed. Assume that Bonnie and Clyde perform the same algorithm in a synchronized manner. Determine and explain the number of communications performed in the worst case. You do not need to analyze the other computations performed, but you may find it helpful to do so.

Hint: Use a divide-and-conquer approach discarding data entries similar to binary search. The number of communications performed should be sublinear ($\Theta(n)$ communications is brute force and not a desired solution.

4) (18 pts.) A car dealership decides to enter the age of Big Data. The manager starts asking questions about daily sales and arising patterns of potential interest.

(i) Your are asked to find an efficient algorithm solving the following sales problem. Given are n integers representing the number of cars sold daily over a period of n days, target value M, and an integer d. Assume the sales are stored in an array of size n. Determine whether there exist at most d consecutive days during which exactly M cars were sold. If one exists, return the first and the last index. Describe and analyze an efficient algorithm for the sales problem. Your solution needs to be better than $O(n^2)$ and $O(dn)$ time.

If more than one solution exists, determine the most recent one (i.e., rightmost index is a maximum). For example, for 3 5 6 0 1 1 1 2 8 12 1 5 2 4 7 8 and $M = 13$ and $d = 4$, the entries $2+4+7 = 13$ represent the solution to be returned.

(ii) Another data set records the daily "3-year warranty yield" results. For every day, the warranty yield entry records the number of customers buying a 3-year warranty minus the number of customer declining a 3-year warranty on a new car. These entries can thus be positive or negative. Given an array of size n representing daily warranty yield entries and a target value Y, determine whether there exists a consecutive sequence of days during which the warranty yield is exactly Y. Describe and analyze an efficient algorithm for the warranty yield problem.