EEC 340/454: Algorithms Spring 2020

Assignment 3: Divide-and-Conquer

Instructors: Mehmet Koyutürk and Orhan Özgüner Due: February 17 before 11:59 PM

Problem 1

Let $0 < \lambda < 1 < a < b$ be constants. Solve the following recurrences using Master Method, noting the case that applies.

(a) $T(n) = bT(n/a) + \Theta(n)$.

(b) $T(n) = a^2T(n/a) + \Theta(n^2)$.

(c) $T(n) = T(\lambda n) + n^\lambda$.

(d) $T(n) = aT(n/a) + \Theta(n^\lambda(\log n)^b)$.

Problem 2

Prove the solutions to the following recurrences using Substitution Method:

(a) For any constant $0 < \alpha < 1$, if $T(n) = T(\alpha n) + T((1-\alpha)n) + \Theta(n)$, then $T(n) = O(n \log n)$.

(b) For any constant $k > 0$, if $T(n) = \Theta(n) + \sum_{i=1}^{k} T(n/2^i)$, then $T(n) = O(n)$.

Problem 3

Dr. Purplesheep discovered a new class of materials, which are useful in converting solar energy to electrical energy. She has designed 25 unique materials that belong to this class, and she would like to identify the top 3 out of these 25 materials in terms of their effectiveness. Unfortunately, however, she does not have an experimental method that can be used to quantify the effectiveness of a given material. Instead, she can run comparative experiments such that each experiment provides a ranking of 5 materials in terms of their effectiveness (from most effective to least effective). Running these experiments is rather expensive, so she would like to minimize the number of experiments she runs. Can you help her design a strategy that will identify the 3 most effective materials among 25 materials by performing the minimum number of experiments? What is the minimum number of experiments needed to conclusively determine the 3 most effective materials? You do not need to write pseudo-code; a verbal explanation, possibly supported by graphics, will suffice.

Example. Suppose Dr. Purplesheep performs an experiment on materials M_3, M_7, M_9, M_{14}, M_{25}. Then, the result of the experiment will be a ranking such as: $M_{14} > M_3 > M_{25} > M_7 > M_9$.

Problem 4

We are given a dictionary D, which is an array of n distinct strings, sorted in lexicographic order. We are also given a procedure COMPARE-STRINGS(x, y), which will compare two strings x and y in $\Theta(1)$ time, and return `true` if x comes before y, and `false` if y comes before x in lexicographic order. We are also given an array C, which contains $n-1$ of the n strings in D, but C is not sorted. We would like to develop an algorithm that will find the string that is missing in C.

(a) Design a divide-and-conquer algorithm that will find the missing string in $\Theta(n)$ time. You can assume that we can utilize the PARTITION algorithm that is used by QUICKSORT, by modifying it to compare strings instead of comparing numbers (you do not need to show how to do the modification). Write the pseudo-code of your algorithm and explain how it works. (Hint: You can select the pivot used for PARTITION algorithm in an informed manner that guarantees the resultant subarrays to be of nearly equal size.)

(b) Show that the runtime of your algorithm is $\Theta(n)$.

Example. Suppose $D = \{"W", "X", "Y", "Z"\}$ and $C = \{"Y", "Z", "W"\}$. D contains $n = 4$ strings and C contains all $n - 1 = 3$ strings in D except "X". Thus, missing string in C is: "X".