CS-473: Algorithms I Spring 2018

Assignment 3: Divide-and-Conquer

Instructor: Mehmet Koyutürk

Due Date and Instructions

Please return hard-copies at the beginning of the class meeting (8:40) on **Friday, March 2, 2018**.

- Hand-writing is accepted but the course personnel reserves the right to reject grading an assignment if the hand-writing is not legible.
- Assignments written in LaTeX will receive 5 bonus points.
- You can submit the assignment as a team of 2 students, however no pair of students are allowed to work together on more than 3 assignments.
- You are also allowed to talk to other students and the course personnel about the solutions, but you must write the answers yourself. Answers copied from other students or resources will be detected, and appropriate action will be taken.

Problem 1

[15 pts.] Dr. Aydin 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.

Problem 2

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

(a) [5 pts] $T(n) = bT(n/a) + \Theta(n)$.

(b) [5 pts] $T(n) = a^2T(n/a) + \Theta(n^2)$.

(c) [5 pts] $T(n) = T(\epsilon n) + n^\epsilon$.

(d) [5 pts] $T(n) = aT(n/a) + \Theta(n^\epsilon(\log n)^b)$.

Problem 3

Prove the solutions to the following recurrences using Substitution Method:

(a) [10 pts] 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) [10 pts] For any constant $k > 0$, if $T(n) = \Theta(n) + \sum_{i=1}^{k} T(n/2^i)$, then $T(n) = O(n)$.

Problem 4

The YouTube video at http://www.youtube.com/watch?v=ywWBy6J5gz8 shows Hungarian Küküllőmenti legényes folk dancers implementing a version of QuickSort. Use this video to answer the following questions:

(a) [10 pts] Write down the pseudo-code for the Partition procedure implemented by the folk dancers.

(a) [10 pts] Prove the correctness of your algorithm using the method of loop invariants and analyze its running time.

Problem 5

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 <u>C is not sorted</u>. We would like to develop an algorithm that will find the string that is missing in C.

(a) [15 pts] 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.

(b) [10 pts] Show that the runtime of your algorithm is $\Theta(n)$.