Habib University
Homework 01
CS 412 (Algorithms: Design and Analysis)

Spring 2024. Total Points: 100

February 21, 2024

1. [10 points] For the following pairs of functions, f and g, justify whether $f = \mathcal{O}(g)$, $f = \Omega(g)$, or both.
   (a) $f(n) = n^n,\ g(n) = n!$
   (b) $f(n) = n \log n,\ g(n) = (\log n)^n$

2. [10 points] Solve the following linear recurrences using the methods discussed in the class. Use the initial conditions to find the coefficients of the general solution.
   (a) $B(n) = 2B(n-1) + \frac{n}{2}$ with $B(0) = 1$.
   (b) $L(n) = L(n-1) + L(n-2)$ with $L(0) = 2$ and $L(1) = 7$.
   (c) $X(n+1) = 2X(n) - X(n-1)$ with $X(0) = 0$ and $X(1) = 1$.
   (d) Show that if $A(n) = 3A(n-1)$ with $A(0) = 1$ then $A(n) = \Theta(3^n)$

3. For each of the divide-and-conquer algorithms described below, give its run time, $T(n)$, as a recurrence relation and then state and justify a tight bound for it.
   (a) [5 points] The solution can be obtained by solving 7 subproblems of size $\frac{n}{7}$ each and then combining the solutions of the subproblems in $(3n + 20)$ steps,
   (b) [5 points] The solution can be obtained by solving 16 subproblems of size $\frac{n}{4}$ each and then combining the solutions of the subproblems in 100 steps,
   (c) [5 points] The solution can be obtained by solving 2 subproblems of size $\frac{n}{2}$ each and then combining the solutions of the subproblems in $(5n^2 + 2n + 3)$ steps.

4. Consider the 1D *peak finding* algorithm as described on pages 2 and 3 here.
   (a) [5 points] Use a recursion tree to state an appropriate asymptotic bound for the recurrence.
   (b) [5 points] Use the substitution method to verify your bound above.

5. [10 points] Given an array A of n integers, design a $\Theta(n)$ algorithm to compute its *prefix sum*, which is defined as the equally-sized array, B, such that

$$B[i] = \sum_{j=1}^{i} A[i], \quad 1 \leq i \leq n.$$

   State and justify the time complexity of your algorithm.

6. [10 points] We define a *local minimum* in a graph as a vertex, v, whose label, x_v, is less than the label, x_u, for all vertices, u, that are adjacent to v. It follows that the local minimum in a graph need not be unique.

   Design an algorithm to find a local minimum (any one) in a complete binary tree containing n nodes. Each node has a distinct numeric label, and the tree can be traversed in the usual manner, i.e., you have direct access to the root, and at any node, you can access its left or right child, if present.

   Your algorithm must run in $O(\log n)$ time. Indicate your algorithm below and justify its running time.

7. [10 points] You are hired by the University as a data analyst to analyze the daily footfall data of students at Habib University, thereby estimating the efficacy of student events happening on campus. Data is available from all the way back till August 2014 when the University welcomed its first batch. An event is considered successful if the footfall on that day is higher than the day of the previous event. The University does not hold more than one event a day. How can you identify the longest run of successful student events? Argue about the complexity of your proposed strategy.

8. [15 points] The ACM International Collegiate Programming Contest (ICPC) is the algorithmic programming World Cup of sorts for college/university students. n teams from Habib have qualified in the preliminary online round and will now travel for the next on-site round. The host offers free registration to the top two teams from each participating institute. Given the unique integer scores of all the Habib University teams in the qualifier, devise a divide-and-conquer algorithm to identify the teams that will not pay the registration fee. Provide pseudocode and the recurrence relation for the running time of your algorithm. Solve the recurrence using any method discussed in the class.

9. [10 points] The Ministry of Magic is faced with a unique challenge. They have received a chest full of n enchanted amulets from the Gringotts Bank, each believed to be linked to some mischievous activities across the realm. These amulets are crafted from ethereal crystals and contain arcane runes that bind them to the vaults of their rightful owners. Due to the complexity of their enchantments, two amulets are considered identical if they unlock the same vault at Gringotts.

   Deciphering the vault number from an amulet's runes is nearly impossible without breaking its magic. However, the Department of Mysteries at the Ministry possesses a powerful artifact known as the 'Arcane Comparator' capable of determining whether two amulets are bound to the same vault without revealing the vault number itself.

   The Gringotts Bank needs to know if there exists a group of more than $\frac{n}{2}$ amulets in their collection, which are all linked to a single vault. Given that the only action the wizards can perform is to compare two amulets using the 'Arcane Comparator', they are tasked with finding a way to answer the bank's question using no more than $\mathcal{O}(nlogn)$ comparisons. Design an algorithm that solves this problem. Determine the recurrences relation and then show that its running time is in $\mathcal{O}(nlogn)$.