[Spring 2022] CSE 317: Design and Analysis of Algorithms [All sections]
Homework 1 | Due: March 10, 2022

Instructions: You should submit it handwritten on A4-size papers. Write your name, section, and ID clearly on top of every page as well as the total number of pages. Direct your all queries to the course staff (refer to syllabus for contact information and office hours).

---

1. (10 points) Show that following algorithm computes $3^n - 2^n$ for all $n \in \mathbb{N} \cup \{0\}$.

   Algorithm: $g(n)$

   1. **if** $n = 0$ **or** $n = 1$ **then return** n
   2. **else return** $(5 \cdot g(n-1) - 6 \cdot g(n-2))$

2. (20 points) Given an array $A[1..n]$ of integers, design an $\Theta(n)$ algorithm to compute *prefix sums*. We define prefix sum for an array A of n integers as an array B of n integers as following:

$$\begin{aligned} B[1] &= A[1] \\ B[2] &= A[1] + A[2] \\ B[3] &= A[1] + A[2] + A[3] \\ &\vdots \\ B[n] &= A[1] + A[2] + \cdots + A[n-1] = \sum_{j=1}^{n} A[j]. \end{aligned}$$

   Prove the time complexity of your algorithm and show it is correct.

3. (30 points) We would like to choose between three divide-and-conquer algorithms for some problem. They are listed below in parts (1), (2), and (3). For each one of them, state the corresponding recurrence relation, and state a tight runtime bound.

   1. If you solve 7 sub-problems of size $n/7$, then the cost of combining the solutions of the subproblems to obtain a solution for the original problem is $3n + 20$,
   2. If you solve 16 subproblems of size $n/4$, then the cost for combining the solutions is 100,
   3. If you solve 2 subproblems of size $n/2$, then the cost for combining the solutions is $5n^2 + 2n + 3$.

4. (20 points) Consider an n-nodes complete binary tree, where $n = 2^d - 1$ for some d. Each node v of T is labeled with a distinct real number x_v (i.e., $\forall u, v \in T, u \neq v \Leftrightarrow x_u \neq x_v$). A node v of T is a *local minimum* if the label x_v is less than the label x_u for all vertices u that are joined to v by an edge.

   You are given such a complete binary T, but the labeling is only specified in the following implicit way: for each node v, you can determine the value x_v by *probing* the vertex v. Show how to find a local minimum of T using only $O(\log n)$ *probes* to the vertices of T.

5. (10 points) Given an array $A[1..n]$ of n numbers (not necessarily distinct). Design an efficient divide and conquer algorithms that finds the *longest increasing contiguous subarray* (LICS) of the array A. For example, if $A = [9, 4, 8, 1, 5, 7, -2, 0]$ then the LICS of A is $[1, 5, 7]$. What is the time complexity of your algorithm?

6. (10 points) Let $G = (V, E)$ be a directed graph with n vertices. A *sink* is a vertex $s \in V$ such that for all $v \in V$, $(v, s) \notin E$. Devise an algorithm that, given the adjacency matrix of G, determines whether or not G has a sink in time $O(n)$.
