Homework 4

CS6033 Design and Analysis of Algorithms I

Fall 2023

(Sec. B, Prof. Yi-Jen Chiang)

Due: Wed. 10/18 by 1pm
(submit online on NYU Brightspace; one submission per group)
Maximum Score: 110 points

Note: This assignment has 2 pages.

1. (15 points)
Textbook Exercise 18.3-1 (page 516). (Note that we use the one-pass top-down deletion algorithm of B-trees as discussed in class and described in the Textbook pages 513-516.) For each deletion, show the tree after each structural change and the final tree, and also state the case number being applied.
(**Note: 5 points for each deletion.**)

2. (16 points)
Recall the following "Baby" Master Theorem that we proved in class:

"Baby" Master Theorem: Let $a > 0, b > 1$ and $d \geq 0$ be constants. Then the solutions of a recurrence of the form

$$T(n) = a\,T(n/b) + \Theta(n^d),$$

where $T(n)$ is a constant for all small enough n, is

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a/b^d < 1, \\ \Theta(n^d \log n) & \text{if } a/b^d = 1, \\ \Theta(n^{\log_b a}) & \text{if } a/b^d > 1. \end{cases}$$

Apply this theorem to solve the recurrences in **(a) - (d)** below, where $T(n)$ is constant for $n \leq 2$. For each recurrence, justify how you apply the theorem (what are the values of a, b, d and which case applies), and express the solution in the $\Theta()$ notation. **(Note: 4 points for each of (a)-(d).)**

(a) $T(n) = 2\,T(n/2) + n^3$.

(b) $T(n) = T(7n/10) + n^2$.

(c) $T(n) = 8\,T(n/4) + n\sqrt{n}$.

(d) $T(n) = 5\,T(n/9) + \sqrt{n}$.

3. (15 points)
Suppose that $T(n)$ is a constant for $n \leq 2$. Solve the following recurrences for $T(n)$ by **repeated unfolding** and express the solutions in the $\Theta()$ notation.

(a) $T(n) = T(n-2) + n^3$. **(7 points)**

(b) $T(n) = 4\,T(n/4) + n \log^2 n$. **(8 points)**

4. (14 points)
Consider the recurrence $T(n)$ given by

$$T(n) = 3\,T(n/3) + \log_3 n,$$

where $T(n) = 0$ when $n = 1$.
Solve $T(n)$ by **changing variables** starting with setting $\log_3 n = m$ and re-writing $T(n)$, followed by **repeated unfolding**. You should derive the **exact expression** for $T(n)$.

5. (10 points)
Consider the deterministic quick selection algorithm (the algorithm SELECT in the Textbook, to find the k-th smallest item from a set of n unsorted items in $O(n)$ worst-case time for any given integer $k \in [1, n]$). Suppose we modify the algorithm so that the items are divided into groups of 7 (rather than groups of 5). Does the algorithm still run in worst-case linear time? Give a new recurrence for the worst-case running time $T(n)$, and solve $T(n)$ and express it in terms of an asymptotic bound, i.e., either show that $T(n) = O(n)$, or show that $T(n) = \omega(n)$.

6. (20 points)
Let $x_1, x_2, \cdots, x_n$ be an unsorted sequence of real numbers (each of which can be positive, negative, or 0), where $n \geq 1$. Design and analyze a **divide-and-conquer** algorithm to find the **subsequence of consecutive elements** such that the product of the numbers in it is minimum over all consecutive subsequences. Here the "subsequence" must have length at least 1; i.e., it must be non-empty. Your algorithm should run in $O(n \log n)$ time.

7. (20 points)
Let T be a tree with root r that is **not necessarily binary**, i.e., each internal node of T can have an **arbitrary** number of children. We say that a path in T is **simple** if there is **no repeated node** on the path. We define the **diameter** of T to be the length of a longest simple path between any two nodes of T, where the **length** of a simple path is the number of edges on that path. Design and analyze a **divide-and-conquer** algorithm to find the diameter of T that runs in worst-case $O(n)$ time where n is the number of nodes in T.