Homework 5

Due: Feb 21 2025

Graded Questions

1. Solve the following recurrences by giving tight Θ-notation bounds in terms of n for sufficiently large n. Assume that $T(\cdot)$ represents the running time of an algorithm, i.e., $T(n)$ is positive and non-decreasing, and for small constants c independent of n, $T(c)$ is also constant. **You need to give the answer and the analysis.** (10 points)

   (a)
   $$T(n) = 9T\left(\frac{n}{5}\right) + n\log n$$

   (b)
   $$T(n) = \sqrt{2025}T\left(\frac{n}{3}\right) + n^{\sqrt{2025}}$$

   (c)
   $$T(n) = 9T\left(\frac{n}{3}\right) + n^2\log n$$

   (d)
   $$T(n) = 10T\left(\frac{n}{2}\right) + 2^n$$

   (e)
   $$T(n) = 3T\left(\frac{n}{4}\right) + n\log^2 n$$

2. Given a square matrix M of size $n \times n$, where each row and each column is sorted in increasing order, design a divide-and-conquer algorithm to find a given value k in M. Assume that n is a power of 2. You need to:

   (a) Describe your algorithm. It **must** be a divide-and-conquer algorithm. No proof of correctness is needed. (5 points)

   (b) Give your algorithm's recurrence relation for runtime complexity. Briefly explain it. (5 points)

   (c) Solve the recurrence relation using the Master Theorem. (5 points)

3. Solve Kleinberg and Tardos, Chapter 5, Exercise 3. You need to:

   (a) Describe your algorithm. It **must** be a divide-and-conquer algorithm. No proof of correctness is needed. (7 points)

   (b) Give your algorithm's recurrence relation for runtime complexity. Briefly explain it. (3 points)

   (c) Solve the recurrence relation using the Master Theorem. (5 points)

4. Emily has received a set of marbles as her birthday gift. She is trying to create a staircase shape with her marbles. A staircase shape contains k marbles in the kth row. Given n as the number of marbles, help her to figure out the number of rows of the largest staircase she can make. (Time complexity $< O(n)$)

   For example, a staircase of size 4 looks like:

$$\begin{array}{llll} * & & & \\ * & * & & \\ * & * & * & \\ * & * & * & * \end{array}$$

   (a) Describe your algorithm. It **must** be a divide-and-conquer algorithm. (5 points)
   (b) Give your algorithm's recurrence relation for runtime complexity. Briefly explain it. (5 points)
   (c) Solve the recurrence relation and state the overall time complexity. (5 points)

Ungraded Questions

1. Solve Kleinberg and Tardos, Chapter 5, Exercise 5.

2. Assume that you have a blackbox that can multiply two integers in one call. Describe an algorithm that, when given an n-bit positive integer exponent a and an integer x, computes x^a using at most $O(n)$ calls to the blackbox.

   You need to:

   (a) Describe your algorithm.
   (b) Provide the recurrence relation for the number of blackbox calls and explain its derivation.
   (c) Solve the recurrence to show that the algorithm uses $O(n)$ calls to the blackbox.