Quick Assignment 3
Total: 100

CS 2500: Algorithms

Due Date: September 6, 2024 at 11.59 PM

Instructions

- Submit your solutions by the deadline specified above.
- Ensure that your work is your own.
- Write your answers clearly and show all your work.
- If you have any questions, ask during recitations or office hours.

Problems

1. In Lecture 6, we guessed that

$$t_n = 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

could be a solution for the recurrence equation

$$t_n = t_{n-1} + n^2$$

Verify that this is indeed the solution using mathematical induction. **[25 points]**

2. In Lecture 6, we guessed that the recurrence equation

$$T(n) = \begin{cases} 0 & \text{if } n = 0 \\ 3T(n \div 2) + n & \text{otherwise} \end{cases}$$

has a solution $T(2^k) = 3^{k+1} - 2^{k+1}$, when $n = 2^k$ and $n > 1$ However, we still need to verify this. Use mathematical induction to verify that the solution $T(2^k) = 3^{k+1} - 2^{k+1}$ holds for all k. **[25 points]**

3. We also showed that the recurrence in part (2) has a general closed-form solution $T(n) = 3n^{\log_2 3} - 2n$, and that $T(n) \in \Theta(n^{\log_2 3})$. This holds because $T(n)$ is a non-decreasing function. Use mathematical induction to prove that $T(n) = 3n^{\log_2 3} - 2n$ is indeed non-decreasing. **[25 points]**

4. Solve the following recurrence equation using the guess-and-verify method and write the complexity in Big-O notation: **[25 points]**

$$T(n) = 2T\left(\frac{n}{2} + 17\right) + n$$