CS 141 homework2

Celyna Su

TOTAL POINTS

86 / 100

QUESTION 1

1 Problem 1 23 / 25

✓ **+ 7 pts Expansion for substitution is correct**

✓ **+ 6 pts Correct result for iterative substitution**

✓ **+ 6 pts Base case proof is correct**

✓ **+ 6 pts Correctly use Inductive steps to prove**

+ **0 pts** Incorrect Answer

+ **3 pts** Prove a bound (correctly) but not the exact solution

✓ **- 2 pts Exact solution is correct but not simplified**

+ **3 pts** Prove the base case but without expanding to show why

QUESTION 2

2 Problem 2 25 / 25

- **5 pts** Case of Master theorem is incorrect

- **5 pts** Conditions are incorrect/not mentioned

- **5 pts** Solution is incorrect

✓ **- 0 pts Correct**

QUESTION 3

3 Problem 3 14 / 25

- **0 pts** Correct

✓ **- 3 pts No working shown for first recurrence**

✓ **- 3 pts No working shown for second recurrence**

✓ **- 3 pts No working shown for third recurrence**

✓ **- 2 pts No/Incorrect choice**

- **2 pts** Incorrect working for first recurrence

- **2 pts** Incorrect working for second recurrence

- **2 pts** Incorrect working for third recurrence

- **1 pts** No/Incorrect T(n) for first recurrence

- **1 pts** No/Incorrect T(n) for second recurrence

- **1 pts** No/Incorrect T(n) for third recurrence

- **1 pts** No/Incorrect T(n) solution for first recurrence

- **1 pts** No/Incorrect T(n) solution for second recurrence

- **1 pts** No/Incorrect T(n) solution for third recurrence

QUESTION 4

4 Problem 4 24 / 25

- **0 pts** Correct

- **2 pts** Incorrect comparison. If ma > mb, choose median from A's first half i.e 0 to n/2 and B's second half i.e n/2 to n. If ma < mb, choose median from A's second half i.e n/2 to n and B's first half i.e 0 to n/2.

- **1 pts** No/Incorrect base case specified.

✓ **- 1 pts No/Incorrect explanation for algorithm being O(log n)**

- **8 pts** Incorrect solution

- **6 pts** Algorithm without explanation.

CS 141, Spring 2019 — Homework 2

Posted: April 11th, 2019 — Due: April 18th, 2019

Name: Celyna Su

Student ID #: 862037643

- You are expected to work on this assignment on your own
- Use pseudocode, Python-like or English to describe your algorithms. Absolutely no C++/C/Java
- When designing an algorithm, you are allowed to use any algorithm or data structure we explained in class, without giving its details, unless the question specifically requires that you give such details
- Always remember to analyze the time complexity of your algorithms
- Homework has to be submitted electronically via Gradescope by the deadline. No late assignments will be accepted

Problem 1. (25 points)

Given the following recurrence relation

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\dfrac{n}{9}\right) + \sqrt{n} & n > 1 \end{cases}$$

1. Solve it exactly (i.e., without using any asymptotic notation) by iterative substitutions
2. Prove by induction that your exact solution is correct (do not prove a bound, but the exact solution)

Answer:

1. $T(\frac{n}{9}) + \sqrt{n}$

2. $\left[T(\frac{n}{81}) + \sqrt{\frac{n}{9}}\right] + \sqrt{n} \longrightarrow T(\frac{n}{81}) + \sqrt{n}(\frac{1}{3} + 1)$

3. $\left[T(\frac{n}{729}) + \sqrt{\frac{n}{81}}\right] + \sqrt{n}(\frac{1}{3} + 1) \longrightarrow T(\frac{n}{729}) + \sqrt{n}(\frac{1}{9} + \frac{1}{3} + 1)$

$\longrightarrow T(\frac{n}{9^i}) + \sqrt{n}(\frac{1}{9} + \frac{1}{3} + 1)$

$\longrightarrow T(\frac{n}{9^i}) + \sqrt{n}(\frac{1}{3^{i-1}} + \frac{1}{3^{i-2}} + ... + \frac{1}{3^0})$

$\longrightarrow T(\frac{n}{9^i}) + \sqrt{n}\sum_{i=0}^{i-1} \frac{1}{3^n}$

$\longrightarrow T(\frac{n}{9^i}) + \sqrt{n}(\frac{1-\frac{1}{3}^i}{1-\frac{1}{3}})$

$\longrightarrow T(\frac{n}{9^i}) + \frac{3}{2}\sqrt{n}(1 - \frac{1}{3}^i)$

$\frac{n}{9^i} = 1$

$9^i = n$

$i = \log_9 n$

4. $T(\frac{n}{9^{\log_9 n}}) + \frac{3}{2}\sqrt{n}(1 - \frac{1}{3}^{\log_9 n})$

5. $T(1) + \frac{3}{2}\sqrt{n}(1 - \frac{1}{\sqrt{n}})$

6. $1 + \frac{3}{2}\sqrt{n}(1 - \frac{1}{\sqrt{n}})$

Proof.

Base Case:

$n = 1$

$T(1) = 1 + \frac{3}{2}(1 - 1)$

$1 + 0 = 1$

$1 = 1$

Inductive Step:

Let $n = \frac{n}{9}$

$T(\frac{n}{9}) = 1 + \frac{3\sqrt{\frac{n}{9}}}{2}(1 - \frac{1}{\sqrt{\frac{n}{9}}})$

$T(\frac{n}{9}) = 1 + \frac{\sqrt{n}}{2}(1 - \frac{3}{\sqrt{n}})$

$T(\frac{n}{9}) = 1 + \frac{\sqrt{n}}{2} - \frac{3\sqrt{n}}{2\sqrt{n}}$

$T(\frac{n}{9}) + \sqrt{n} = 1 + \frac{\sqrt{n}}{2} - \frac{3}{2} + \sqrt{n}$

$\longrightarrow 1 + \frac{3\sqrt{n}}{2} - \frac{3}{2} = 1 + \frac{3}{2}\sqrt{n}(1 - \frac{1}{\sqrt{n}})$ **(from solution at 6)**

Problem 2. (25 points)

Using the Master method, give an asymptotic tight bound for $T(n)$ in the following recurrence relation

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\frac{n}{3}\right) + n\log_3 n & n > 1 \end{cases}$$

Answer:

$a = 1, b = 3, f(n) = n\log_3 n$

For this, $n^{\log_b a} = n^{log_3 1} = n^0 \longrightarrow 1$

(Case 3)

For $f(n)$ is $\Omega(n^{0+\epsilon})$, for $\epsilon = 1\delta = \frac{1}{3}$, and $af(\frac{n}{b}) = \frac{n}{3}\log_3(\frac{n}{3})$

$\longrightarrow \frac{n}{3} * \log_3 n - \log_3 3 = \frac{n}{3} * \log_3 n - 1$

$\frac{f(n)}{3} - 1 < \frac{1}{3}f(n)$

$\theta(n\log_3 n)$

Problem 3. (25 points)

Suppose that we have designed three divide-and-conquer algorithms that solve a particular problem, where the input size is n. The first one solves four subproblems of size $n/2$ and the cost of combining the solutions of the subproblems to obtain a solution for the original problem is n^2. The second solves three subproblems of size $n/2$ and requires $n^2\sqrt{n}$ time for combining the solutions. The third solves five subproblems of size $n/2$ and requires $n \log n$ time for combining the solutions. Assume that all three take $\Theta(1)$ when $n = 1$. Which algorithm would you choose and why? Show your work using the Master method.

Answer:

Algorithm 1:

$a = 4, b = 2, f(n) = n^2$

$$T(n) = \begin{cases} 1 & n = 1 \\ 4T(\frac{n}{2}) + n^2 & n > 1 \end{cases}$$

Algorithm 2:

$a = 3, b = 2, f(n) = n^2\sqrt{n}$

$$T(n) = \begin{cases} 1 & n = 1 \\ 3T(\frac{n}{2}) + n^2\sqrt{n} & n > 1 \end{cases}$$

Algorithm 3:

$a = 5, b = 2, f(n) = n \log n$

$$T(n) = \begin{cases} 1 & n = 1 \\ 5T(\frac{n}{2}) + n & n > 1 \end{cases}$$

Problem 4. (25 points)

The *median* of a set of numbers $\{a_1, a_2, \ldots, a_n\}$ is the element a_i such that there are $\lceil n/2 \rceil$ elements smaller than or equal to a_i, and there are $\lfloor n/2 \rfloor$ greater than or equal to a_i. In other words, the median is the element in the middle when the elements are sorted. For example, the median of $\{7, 3, 4, 1, 9, 2, 13\}$ is 4.

You are given two sorted arrays A and B of size n each (for simplicity, you can assume n to be some power of 2 and that the numbers are distinct). Give an algorithm to find the median of all $2n$ numbers in $O(\log n)$ time.

Answer:

1. Find the medians i, j of the two sorted arrays A and B.
2. If $i = j$: the median is found.
3. If $i > j$: the median is found in the first element of A to i or j to the last element of B.
4. If $j > i$: the median is found in i to the last element of A or the first element of B to j.
5. Repeat above steps until the size of A and B are 2.
6. Now, median = (max(A[0], B[0]) + min(A[1], B[1]))/2