Ayça Elif Aktaş - 27802

Sabancı University
Faculty of Engineering and Natural Sciences

CS301 – Algorithms

Homework 1

Due: March 6, 2024 @ 23.55
(upload to SUCourse)

PLEASE NOTE:

- Provide only the requested information and nothing more. Unreadable, unintelligible, and irrelevant answers will not be considered.
- Submit only a PDF file. (-20 pts penalty for any other format)
- Not every question of this homework will be graded. We will announce the question(s) that will be graded after the submission.
- You can collaborate with your `TA/INSTRUCTOR ONLY` and discuss the solutions of the problems. However, you have to write down the solutions on your own.
- Plagiarism will not be tolerated.

Late Submission Policy:

- Your homework grade will be decided by multiplying what you normally get from your answers by a "submission time factor (STF)".
- If you submit on time (i.e. before the deadline), your STF is 1. So, you don't lose anything.
- If you submit late, you will lose 0.01 of your STF for every 5 mins of delay.
- We will not accept any homework later than 500 mins after the deadline.
- SUCourse's timestamp will be used for STF computation.
- If you submit multiple times, the last submission time will be used.

Question 1

The recurrence relation of a recursive divide and conquer algorithm is given. Explain this recurrence, verbally, in terms of the size of each sub-problem, the cost of dividing the problem, and combining solutions.

$$T(n) = 3T(\tfrac{n}{4}) + 2n + n^3$$

Answer:

→ we have 3 sub problems

→ Each sub problem have quarter size of the original problem. $n/4$

→ It takes $2n$ time to divide the problem into sub problems.

→ It takes $C(n) = n^3$ time to combine the solutions from the sub problems.

Question 2

Find an asymptotically tight lower bound for the following recurrence by using the substitution method.

$$T(n) = T(\tfrac{n}{3}) + T(\tfrac{2n}{3}) + \Theta(n)$$

Answer:

→ Make a guess for the lower bound

$$T(n) = \Omega(n \lg n)$$

→ Claim hypothesis based on my guess

$T(n) \geq c n \lg n$ • we assume that there exist a constant $c > 0$ that satisfies this equation

→ Show hypothesis to be true with substitution method

- Lets substitute $T(n)$ into the recurrence
- $T(n) \geq T(\frac{n}{3}) + T(\frac{2n}{3}) + \Theta(n)$
- Assuming that $T(n) \geq c n \lg n$

$$c n \lg n \geq c(\tfrac{n}{3})\lg(\tfrac{n}{3}) + c(\tfrac{2n}{3})\lg(\tfrac{2n}{3}) + \Theta(n)$$

- lets simplify right hand side

$$= c(\tfrac{n}{3})\lg n - c\tfrac{n}{3}\lg 3 + c(\tfrac{2n}{3})\lg n - c\tfrac{2n}{3}\lg 3 + \Theta(n)$$

$$= c n \lg n - (\tfrac{c}{3} n \lg 3 - \tfrac{2c}{3} n \lg 3) + \Theta(n)$$

$$= c n \lg n - c n \lg 3 + \Theta(n)$$

we can choose c as $c \leq \frac{1}{\lg 3}$ to ensure that $T(n) \geq c n \lg n$.
Now we proved that $T(n) \leq c n \lg n$ satisfies with $c \leq 1/\lg 3$ so we accept our initial assumption.

$\boxed{\text{Answer} = \Omega(n \lg n)}$

Question 3

For the following recurrences, either solve it by using the master method or show that it cannot be solved with the master method.

(a) $T(n) = T(\frac{n}{2}) + \Theta(1)$

Answer:

$\rightarrow$ $f(n) = \Theta(1)$

$\rightarrow$ $n^{\log_{b \to 2}^{a \to 1}} = n^{\log_2 1} = n^0 = 1$

$\rightarrow$ It indicates that $\underbrace{f(n)}_{\Theta(1)} = \Theta(\underbrace{n^{\log_2 1}}_{1})$

$\rightarrow$ then this follows MT case 2 $\boxed{T(n) = \Theta(\log n)}$

(b) $T(n) = 3T(\frac{n}{4}) + n \lg n$

Answer:

$\rightarrow$ $f(n) = n \log n$

$\rightarrow$ $n^{\log_4 3}$ it is between 0 and 1

$\rightarrow$ $n^{\log_4 3} = O(n \log n)$

$\rightarrow$ $f(n) = \Omega(n^{\log_4 3})$, this follows MT case 3

check: $a f(n/b) \leq c f(n)$, $c < 1$

$3 \frac{1}{4} \log n/4 \leq c n \log n$ $\quad c = \frac{3}{4}$ satisfies this condition

$\boxed{T(n) = \Theta(f(n)) = \Theta(n \log n)}$