Data Structures and Algorithms

Natalia Martinez
9/5/2023

Homework 1

Due: 9/7/2023 on Canvas
(100 points)

Question 1 (30 points) Answer the following questions. All running times refer to the worst-case analysis.

1) Abby came up with an algorithm for some problem that runs in time $\Theta(n^2 \log(n))$, and Bill came up with an algorithm for the same problem that runs in time $o(n^2 \log(n))$. Based on this information, which one would you choose? Why?
    a. Based on this information, I would choose Bill's algorithm since o has a tighter upper bound than theta.

2) Abby tells you that a certain algorithm runs in time $O(n^3 + 200n)$, and Bill tells you that the same algorithm runs in time $\Omega(n^3)$. Can both Abby and Bill be correct? Why?
    a. Yes, both Abby and Bill can be correct because big oh gives an upper bound, and omega gives a lower bound.

3) Abby tells you that a certain algorithm runs in time $\Omega(n^2 + 200n)$, and Bill tells you that the same algorithm runs in time $\Omega(n^3)$. Assume that both statements are correct, which one is more informative, i.e., gives you a better estimation of the running time? Why?
    a. In this case Bill's would be more informative since it gives a tighter bound than Abby's.

Question 2 (25 points) Show that $\frac{n^2}{2} - 2n = \Theta(n^2)$. Show your work and give specific values for c_1, c_2, and n_0.

$$c_1 \times n^2 < \frac{n^2}{2} - 2n \le c_2 \times n^2 \quad \text{for all } n \ge n_0$$

$$\downarrow$$

$$c_1 \le \frac{1}{2} - \frac{2}{n} \le \frac{c_2}{n}$$

- As $n \to \infty$ $\quad c_2 = \frac{1}{2}$

$$c_1 \times n^2 \le \frac{n^2}{2} - 2n \qquad c_1 = \frac{1}{4} \quad n_0 = 4$$

$$2 \cdot \frac{1}{4} n^2 \le \left(\frac{n^2}{\cancel{2}} - 2n\right) \cdot 2$$

$$2 \cdot \frac{1}{2} n^2 \le (n^2 - 4n)\, 2$$

$$n^2 \le n^2 - 8n \quad \text{true for } n \ge 0$$

$c_2 = 1 \quad n_0 = 1 \quad n \geq 1?$

$\frac{n^2}{2} - 2n \leq \frac{1}{2}n^2 \Rightarrow \therefore \frac{n^2}{2} - 2n = \Theta(n^2)$ when $c_1 = \frac{1}{4}$

$c_2 = \frac{1}{2}$

$n_0 = 4$

Question 3 (25 points) Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions (a function is asymptotically nonnegative whenever n is sufficiently large). Show that $f(n) + g(n) = O(max(f(n), g(n)))$. Show your work and give specific values for c and n_0.

Hint: For a given n, $max(f(n), g(n))$ is $f(n)$ if $f(n) \geq g(n)$, otherwise it is $g(n)$.

Question 4 (20 points) Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$. Do not use the Master Theorem.

#3) $c = 2 \quad n_0 = K_1$ For $n \geq n_0$

$|f(n) + g(n)| \leq f(n) + |g(n)| \leq f(n) + g(n)$

since $f(n) \geq g(n)$

$+$

$f(n) + g(n) \leq 2f(n)$

$\therefore$ for $n \geq n_0 \rightarrow |f(n) + g(n)| \leq c \cdot max(f(n), g(n))$ when $f(n) \geq g(n)$

$+$

ineq: $|f(n) + g(n)| \leq c \cdot g(n)$

$c = 2 \quad n_0 = k_2 \quad$ For $n \geq n_0$

$|f(n) + g(n)| \leq |f(n)| + g(n) \leq g(n) + g(n)$

Since $f(n) < g(n) \rightarrow f(n) + g(n) \leq 2g(n)$

$\therefore n \geq n_0$ shows $|f(n) + g(n)| \leq c \cdot \max(f(n), g(n))$ when $f(n) \leq g(n)$

$\therefore$ it can be concluded that $f(n) + g(n) = O(\max(f(n), g(n)))$

#4) $T(n) = T(n-1) + n$ is $O(n^2)$ $\quad T(1) = 1$

$T(n) = an^2 + bn + c$

$T(1) = a(1)^2 + b(1) + c$

$1 = a + b + c$ $\quad$ // $T(k) <= ak^2 + bk + c$

$T(n) <= an^2 + bn + c$

$T(n) = T(n-1) + n$ → substitute into eq

$T(n) <= a(n-1)^2 + b(n-1) + c + n$ → expand

$(n-1)(n-1)$

$a(n^2 - 1n - 1n + 1)$

$an^2 - 2an + a + bn - b + c + n$

$T(n) <= an^2 + (b-2a)n + (a - b + c) + n$

$a = 1/2$

$b - 2a = 0 \rightarrow b = 1$

$a - b + c = 0 \quad c = 1/2$

$T(n) <= \frac{1}{2}n^2 + \cancel{(1 - 2\frac{1}{2})n} + (\frac{1}{2} - 1 + \frac{1}{2}) + n$

$\frac{1}{2}n^2 + \frac{1}{2}\cancel{n - n + \frac{1}{2}n}$

$T(n) <= \frac{1}{2}n^2 \quad \therefore T(n) <= O(n^2)$