Algorithms and Data Structures Coursework

Finlay Boyle

November 2019

Question 4

Part a

If $f(x) + g(x)$ is $o(f(x)g(x))$ then $\lim_{x\to\infty} \frac{f(x)+g(x)}{f(x)g(x)} = 0$

$$\begin{aligned}\lim_{x\to\infty} \frac{f(x)+g(x)}{f(x)g(x)} &= \lim_{x\to\infty} \left(\frac{f(x)}{f(x)g(x)} + \frac{g(x)}{f(x)g(x)}\right) \\ &= \lim_{x\to\infty} \left(\frac{1}{g(x)} + \frac{1}{f(x)}\right)\end{aligned}$$

This is not true for all asymptotically positive functions. A counter-example to this is $f(x) = \frac{x}{x+2}$ and $g(x) = x^4$. $f(x)$ is asymptotically positive since $x^4 > 0 \ \forall x \neq 0$ and $g(x)$ is asymptotically positive since $\frac{x}{x+2} > 0 \ \forall x > 0$.

$$\lim_{x\to\infty} \frac{1}{f(x)} = \lim_{x\to\infty} \frac{1}{x^4} = 0$$

and

$$\begin{aligned}\lim_{x\to\infty} \frac{1}{g(x)} &= \lim_{x\to\infty} \frac{1}{\frac{x}{x+2}} \\ &= \lim_{x\to\infty} \frac{x+2}{x} \\ &= \lim_{x\to\infty} \left(1 + \frac{2}{x}\right) \\ &= 1\end{aligned}$$

hence

$$\lim_{x\to\infty} \left(\frac{1}{g(x)} + \frac{1}{f(x)}\right) = 1 + 0 = 1$$

So these functions are a counter-example so $x^4 + \frac{x}{x+2}$ is not $o(x^4 * \frac{x}{x+2})$ hence it is **false**.

Part b

If $2^x x^2$ is $o(2.1^x)$ then $\lim_{x\to\infty} \frac{2^x x^2}{2.1^x} = 0$

$$\begin{aligned}\lim_{x\to\infty} \frac{2^x x^2}{2.1^x} &= \lim_{x\to\infty} x^2 \left(\frac{2}{2.1}\right)^x \\ &= \lim_{x\to\infty} x^2 \left(\frac{2.1}{2}\right)^{-x} \\ &= \lim_{x\to\infty} \frac{x^2}{\left(\frac{2.1}{2}\right)^x}\end{aligned}$$

Then since the exponential is monotonic increasing as its base is greater than 1 and exponentials beat powers as $x \to \infty$ then $\lim_{x\to\infty} \frac{2^x x^2}{2.1^x} = 0$ hence $2^x x^2$ is $o(2.1^x)$ is **true**.

Part c

f(x) = $x^2 log(x)$, g(x) = x^2

Assume $\exists$ $C \in (0,\infty)$ and $k \in (0,\infty)$ such that $x^2 log(x) \le Cx^2$ $\forall$x $\ge$ k then $log(x) \le C$. However, $log(x)$ is monotonic increasing so $log(x) > C$ for sufficiently large x hence there does not exists a witness pair C, k which is a contradiction so $x^2 log(x)$ is not $O(x^2)$, the statement is **false**.

Part d

f(x) = $x^2 log(x)$, g(x) = x^3 then

$$\begin{aligned}x^2 log(x) &\le Cx^3 \\ log(x) &\le Cx\end{aligned}$$

This is **true** for x $\ge$ 1 as $log(1) = 0$ and $x = 1$, x grows faster than $log(x)$ therefore giving the witness pair C = 1, k = 1 so $x^2 log(x)$ is $O(x^3)$

Part e

f(x) = $7x^5$, g(x) = $12x^4 + 5x^3 + 8$

Assume $\exists$ $C \in (0,\infty)$ and $k \in (0,\infty)$ such that $7x^5 \le C(12x^4 + 5x^3 + 8)$ $\forall$x $\ge$ k then:

$$\begin{aligned}7x^5 &\le C(12x^4 + 5x^3 + 8) \\ 7x &\le 12C + \frac{5C}{x} + \frac{8C}{x^4}\end{aligned}$$

As $x \to \infty$, $\frac{1}{x} \to 0$ and $\frac{1}{x^4} \to 0$ so:

$$\lim_{x\to\infty} \frac{5C}{x} = 0, \lim_{x\to\infty} \frac{8C}{x^4} = 0$$

So for sufficiently large x, $\frac{5C}{x}$ and $\frac{8C}{x^4}$ become insignificant so the RHS is a decreasing function whereas the LHS is a monotonic increasing function hence $7x^5 > C(12x^4 + 5x^3 + 8)$ for sufficiently large x so there cannot exist a witness pair C, k which is a contradiction therefore $7x^5$ is not $O(12x^4 + 5x^3 + 8)$ hence the statement is **false**.

Question 5

Part a

(Can f(n) be negative??) $T(n) = 64T(\frac{n}{8}) - n^2log(n)$ so $a = 64$, $b = 8$, $f(n) = -n^2log(n)$
Since a and b are constants the Master Theorem will apply if the recurrence satisfies one of the cases:

$$n^{log_b a} = n^{log_8 64} = n^2$$
$$f(n) = -n^2log(n) = \Theta(n^2log^1(n))$$

It is clear that $f(n) = \Theta(n^2log^1(n))$ because it only differs by a constant of -1 so the recurrence satisfies $\Theta(n^2log^1(n))$ hence the Master Theorem can be applied since $k \geq 0$:

$$T(n) = \Theta(n^{log_b a}log^k(n)) = \Theta(n^2log^2(n))$$

Therefore, $T(n) = \Theta(n^2log^2(n))$

Part b

$T(n) = 4T(\frac{n}{2}) + \frac{n}{log(n)}$ so $a = 4$, $b = 2$ and $f(n) = \frac{n}{log(n)} = nlog^{-1}(n)$
The Master Theorem may apply here because a and b are both constants.

$$n^{log_b a} = n^{log_2 4} = n^2$$

This appears to be case 1 of the Master Theorem because for case 2, $log^k(n)$ requires $k \geq 0$ whereas $k = -1$ here.
To use case 1, $f(n) = O(n^{2-\epsilon})$:

$$\frac{n}{log(n)} \leq Cn^{2-\epsilon}$$
$$\frac{1}{log(n)} \leq Cn^{1-\epsilon}$$

Since $n^{1-\epsilon}$ is monotonic increasing and tends towards ∞ because ϵ is a small positive constant and $log(n)$ is also monotonic increasing as n $\to \infty$ so $\frac{1}{log(n)} \to 0$. So the inequality holds for $n \geq 2$, $C = 1$ (assuming $log(n) = log_2(n)$). Hence,

$$f(n) = O(n^{2-\epsilon})$$

The Master Theorem therefore applies so:

$$T(n) = \Theta(n^{log_b(a)}) = \Theta(n^2)$$

Part c

$T(n) = 2^nT(\frac{n}{2}) + n^n$ so $a = 2^n$, $b = 2$ and $f(n) = n^n$
The Master Theorem cannot be applied here because a is not a constant which is required for the Master Theorem. Hence, $T(n)$ cannot be solved by the Master Theorem.

Part d

$T(n) = 3T(\frac{n}{4}) + nlog(n)$ so $a = 3$, $b = 4$ and $f(n) = nlog(n)$
As a and b are constants, the Master Theorem may apply here if it satisfies one of the cases.

$$n^{log_b(a)} = n^{log_4(3)} \approx n^{0.792}$$

Since $O(n^{log_4(3)}) < O(n) < O(nlog(n))$, this appears to case 3 of the Master Theorem.
To use case 3, $f(n) = \Omega(n^{log_4(3)})$

$$\begin{aligned} Cnlog(n) &\geq n^{log_4 3} \\ Clog(n) &\geq n^{log_4 3 - 1} \\ Clog(n) &\geq \frac{1}{n^{1-log_4 3}} \end{aligned}$$

This inequality holds for $C = 1$ and $n \geq 2$ (assuming $log(n) = log_2(n)$) since $Clog(n)$ and $n^{1-log_4 3}$ are both monotonic increasing so $\frac{1}{n^{1-log_4 3}} \to 0$ as $n \to \infty$. Hence,

$$f(n) = \Omega(n^{log_4(3)})$$

Then for the second condition, $af(\frac{n}{b}) \leq cf(n)$ for some $c < 1$ and $n \geq k$:

$$\begin{aligned} 3f\left(\frac{n}{4}\right) &\leq cf(n) \\ \frac{3n}{4} log\left(\frac{n}{4}\right) &\leq cnlog(n) \\ \frac{3}{4}(log(n) - log(4)) &\leq clog(n) \\ \frac{3}{4}log(n) - \frac{3}{4}log(4) &\leq clog(n) \end{aligned}$$

So for sufficiently large n, if $\frac{3}{4} \leq c < 1$ then this holds true as $\frac{3}{4}log(4)$ becomes insignificant as it is a constant. Hence, $\exists c < 1$ such that $af(\frac{n}{b}) \leq cf(n)$ is true so this is case 3 of the Master Theorem:

$$T(n) = \Theta(f(n)) = \Theta(nlog(n))$$

Part e

$T(n) = 3T(\frac{n}{3}) + \sqrt{n}$ so $a = 3$, $b = 3$ and $f(n) = \sqrt{n}$
As a and b are constants, the Master Theorem may apply here if it satisfies one of the cases.

$$n^{log_b(a)} = n^{log_3(3)} = n$$

This is clearly case 1 so $f(n) = O(n^{log_b(a)-\epsilon})$:

$$\begin{aligned} n^{0.5} &\leq Cn^{1-\epsilon} \\ 1 &\leq Cn^{0.5-\epsilon} \end{aligned}$$

This is true for $C = 1$, $n \geq 1$ and $\epsilon < 0.5$ (but ϵ is a small positive constant anyway). Hence, $f(n) = O(n^{log_b(a)-\epsilon})$ thus

$$T(n) = \Theta(n^{log_b(a)}) = \Theta(n)$$

Question 6

Part b

The worse-case input for this algorithm for k pivots is:

- The first 2k elements of the array are reverse-sorted.
- The next k elements are all are greater than the first 2k elements. These elements are also reverse sorted.
- Repeat this for as many k element blocks each one must contain elements greater than all previous elements in the array. The k elements must be reverse-sorted.

For example, for a 15-element array with k = 5 a worst-case input would be [10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 15, 14, 13, 12, 11, 20, 19, 18, 17, 16].

This is the worst-case input because when the algorithm selects the pivots it will select them from the right end. It will pick k elements which are reverse-sorted and then it will sort them using insertion sort. The worst-case for insertion sort is a reverse-sorted array. Then it will partition the array between these pivots however none of the elements in the array will be put in the partitions producing a n - k partition and k-sorted elements on the right side of the array. It will repeat this until there are 2k elements left. Each time it only reduces the problem by k-elements.

Once there are 2k elements left it will sort them using insertion sort. These elements will are also reverse-sorted which is again the worst-case for insertion sort. This overall gives the worst-case.