Compsci 220 A1 Joanne Chen upi: jehc820

1. (a) ① $\lim\limits_{n\to\infty} \frac{T_2}{T_1} = \frac{10^n + 10^8}{10^8 \cdot n^4 + n^8} = \infty$

Because $T_2(n) = 10^n + 10^8$ is Exponential plus constant, which obviously grows faster than n^4 and n^8

$\therefore$ T_2 grows faster than T_1, So A_2 is slower than A_1.

② $\lim\limits_{n\to\infty} \frac{T_3}{T_2} = \lim\limits_{n\to\infty} \frac{10^n + \log_{10}(n)}{10^n + 10^8} = \lim\limits_{n\to\infty} \frac{10^n}{10^n + 10^8} + \lim\limits_{n\to\infty} \frac{\log_{10}(n)}{10^n + 10^8}$

$= \lim\limits_{n\to\infty} \frac{10^n}{10^n + 10^8} + 0 = 1$

But T_3 is Exponential plus Logarithmic, while T_2 is Exponential plus constant, hence, T_3 grows faster than T_2, so A_3 is slower than A_2

③ $\lim\limits_{n\to\infty} \frac{T_3}{T_1} = \lim\limits_{n\to\infty} \frac{10^n + \log_{10}(n)}{10^8 \cdot n^4 + n^8} = \lim\limits_{n\to\infty} \frac{10^n}{10^8 \cdot n^4 + n^8} + \lim\limits_{n\to\infty} \frac{\log_{10}(n)}{10^8 \cdot n^4 + n^8}$

$= \infty + 0 = \infty$

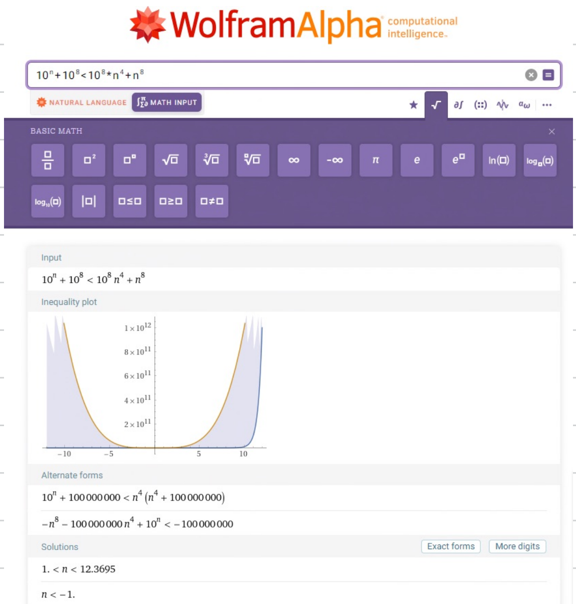
Therefore, T_3 grows faster than T_1. Hence, A_3 is slower than A_1.

To conclude, A_1 is the fastest algorithm, A_3 is the slowest one for the very large inputs.

(b) When A_2 is faster than A_1, $T_2 < T_1$:

$10^n + 10^8 < 10^8 \cdot n^4 + n^8$

$\Rightarrow$ When $n < -1$ or $1 < n < 12.37$



Because $n \in \mathbb{N}$ in this question, for all values of n in the range $2 \le n \le 12$, A_2 is faster than A_1.

(c) When A_3 faster than A_2, $T_3 < T_2$:

$10^n + \log_{10}(n) < 10^n + 10^8$

$\log_{10}(n) < 10^8$

$\Rightarrow 0 < n < 10^{10^8}$ $\Rightarrow$ For $\forall n \in \mathbb{N}$ and $0 < n < 10^{10^8}$ A_3 is faster than A_2.

2. For the worst case, $n \geq 9$ (condition $n < 9$ is false)

Line 2: 1 comparison between n and 9 $\Rightarrow$ 1 elementary operation

Line 4: this for loop includes 2 operations for each iteration:

① variable assignment: $i \leftarrow 1, i \leftarrow 2, \dots i \leftarrow n$

② comparison: $i \leq n$ ③ extra $i \leftarrow n+1$ and comparison $i = n+1 \leq n$ or not?

Line 5: 1 variable assignment for each iteration

Thus, it costs $2+1 = 3$ operations from line 4 to 5 for each iteration.

There are totally n iterations. Hence, total operations from line 4 to 5 are $3n+2$ operations.

Line 6: this for loop includes $i \leftarrow 9$ (1 assignment) and $i \leq n$ (1 comparison)

2 operations for each iteration. 2 extra operation at $i \leftarrow n+1$, $i \geq n$

Line 7: For each iteration it costs 2 operations:

1 arithmetic operations $(i-8)$

1 comparison (compare $b[i-8] = 0$ or $\neq 0$)

Line 6	: i	9	10	11	12	13	14	15	16	17
line 7	$i-8$	1	2	3	4	5	6	7	8	9
	$b[i-8]$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$= 0$
Line 8	return	×	×	×	×	×	×	×	×	✓
Line 9	$b[i] \leftarrow 0$	$b[9] \leftarrow 0$	$b[10] \leftarrow 0$			-----			$b[16] \leftarrow 0$	Not execute

so it costs 1 more operation $b[i] \leftarrow 0$ for each iteration when $n \geq 17$ and $i < 17$

For line 6 ~ 9 using worst case, it costs totally $2+2+1 = 5$ operations for each iteration. and for worst case, line 7 runs 8 iterations, when $i = 17$, $b[9]$ will be zero as it has been assigned to be zero when $i = 9$, $i-8 = 1$, $b[1] \neq 0$.

Then when $i = 17$, line 8 return $i = 17$

Hence, when $9 \leq n \leq 16$, there are $n-9+1 = n-8$ iterations,

total operations from line 6 to 9 are $5(n-9+1)+2 = 5(n-8)+2$ operations.

When $n > 16$, line 6 ~ 9 have $8 \times 5 + 4 = 44$ operations

Line 10: function call and arithmetic operations $(n-1)$ $\Rightarrow$ $2 + T(n-1)$ operations

To conclude, if $n < 9$, $T(n) = 1$

if $9 \leq n \leq 16$, $T(n) = 1 + (3n+2) + 5(n-8) + 2 + 2 + T(n-1) = 8n - 33 + T(n-1)$

if $n > 16$, $T(n) = 1 + 3n + 2 + 44 = 3n + 47$

3. a) $\lim_{n\to\infty} \frac{\log_2(n^8)}{\log_2(n^4)} = \frac{8\log_2(n)}{4(\log_2(n))} = 2$

Because using limit rule, $0 < 2 < \infty$, then $\log_2(n^8)$ is $\Theta(\log_2(n^4))$

$\therefore$ $\log_2(n^8)$ is $O(\log_2(n^4))$

let $C=2$, $n_0=1$, then

$\log_2(n^8) = 8\log_2(n) = 2\log_2(n^4)$

$\therefore \forall n \geq 1$, $\log_2(n^8) \leq 2\log_2(n^4)$ $\therefore \forall n \geq n_0$, $\log_2(n^8) \leq C \cdot \log_2(n^4)$

$\therefore$ $\log_2(n^8)$ is $O(\log_2(n^4))$

b) Let $C=1$, $n_0=2$, then

if $n=2$, then $2^8 > 2^4$

$\lim_{n\to\infty} \frac{n^8}{n^4} = \lim_{n\to\infty} n^4 = \infty$

$\therefore$ Using limit rule, n^8 is not $O(n^4)$

c) $-1 \leq \cos(n) \leq 1 \Rightarrow -1+2 \leq \cos(n)+2 \leq 1+2$

$1 \leq \cos(n)+2 \leq 3$

$\therefore$ when n is a very large value, $\frac{1}{10} \leq \frac{\cos(n)+2}{10} \leq \frac{3}{10}$

$\therefore$ $\cos(n)+2$ is $\Theta(10)$

Let $C=12$, $n_0=3$, then $12 \leq 12(\cos(n)+2) \leq 26$
then $10 \leq 12(\cos(n)+2))$ for all $n \geq n_0$

$\therefore 10$ is $O(\cos(n)+2)$ $\therefore$ $\cos(n)+2$ is $\Omega(10)$

d) $-1 \leq \cos(n) \leq 1 \Rightarrow 0 \leq \cos(n)+1 \leq 2$
Assume $\cos(n)+1$ is $\Omega(10)$, then $\lim_{n\to\infty} \frac{\cos(n)+1}{10} = \infty$ and 10 is $O(\cos(n)+1)$

For any constant C, $0 \leq C\cdot(\cos(n)+1) \leq 2C$, which means that
$C(\cos(n)+1)$ might be less than 10 because $C\cdot(\cos(n)+1)$ could be zero.

$\therefore$ 10 is not $O(\cos(n)+1)$ and $\cos(n)+1$ is not $\Omega(10)$

e) $s(n) = p(n) \cdot q(n) + p(n) \cdot r(n)$

Because p is in $\Theta(2^n) \Rightarrow p$ is in $O(2^n)$ and $\Omega(2^n)$

Using product rule, $p(n)$ is $O(2^n)$, $q(n)$ is $O(3^n)$, then $p(n) \cdot q(n)$ is $O(2^n \cdot 3^n)$

Similarly, $r(n)$ is $O(4^n)$, then $p(n) \cdot r(n)$ is $O(2^n \cdot 4^n)$

Using sum rule, $p(n) \cdot q(n) + p(n) \cdot r(n)$ is in $O(\max\{2^n \cdot 3^n, 2^n \cdot 4^n\})$

As $2^n \cdot 4^n > 2^n \cdot 3^n$ for all $n > 0$

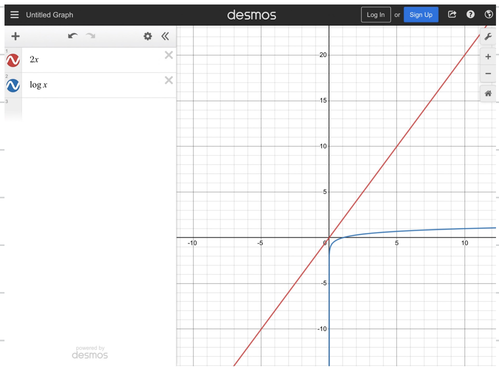
Hence $p(n) \cdot q(n) + p(n) \cdot r(n)$ is in $O(2^n \cdot 4^n)$

4. Line 2 to 4:

Line 2: $i \leftarrow 1$ to $2n \Rightarrow 2n$ iterations

Line 3: $j \leftarrow i$ to $\log n$

i	1	2	3	...	$\log n$
j	1 to $\log n$	2 to $\log n$	3 to $\log n$	...	$\log n$



However, when $i > \log n$, line 3 does not execute

so when $j = i \leq \log n$, inner loop executes

Therefore,

the maximum value of j will be $\log n$ (n is $2^{2k} \Rightarrow \log n = \log_2(2^{2k}) = 2k$ is integer)

The total number of iteration of line 3 inner loops are:

$\log n + \log n - 1 + \log n - 2 + \ldots + 2 + 1 = \frac{1}{2} \log n \cdot (\log n + 1) = \frac{\log n(\log n+1)}{2}$

Hence, line 4 has $C \cdot \frac{\log n(\log n+1)}{2}$ operations.

Line 5 to 10:

line 5: n iterations

Line 6:	i	1	2	3	4	5	6	7	8	9	...	$n = 2^{2k}$
	perfect square	1^2	$\times$	$\times$	2^2	$\times$	$\times$	$\times$	$\times$	3^2	...	$\sqrt{n} = 2^k$

Line 7: C execute $C \cdot \sqrt{n}$ times (Because $n = 2^{2k}$, $\sqrt{n} = 2^k$, $\sqrt{n} \in \mathbb{N}$)

Line 9:	j	$\times$	$1 \sim n$	$1 \sim n$	$\times$	$1 \sim n$	$1 \sim n$	$1 \sim n$	$1 \sim n$	$\times$	... $\times$

Line 10: C execute $C \cdot (n - \sqrt{n}) \cdot n$ times

Line 12 to 14:

line 12:	i	$2 = 2^1$	$4 = 2^2$	$16 = 2^4$	2^8	...	$\sqrt{n} = 2^k$	$n = 2^{2k}$
Line 13:	$i \leftarrow i \cdot i$	$4 = 2^2 = 2^{2^1}$	$16 = 2^4 = 2^{2^2}$	$256 = 2^8 = 2^{2^3}$	$2^{16} = 2^{2^4}$	...	$n = 2^{2^a} = 2^{2k}$	not execute

Line 14: C execute $a \cdot C$ times $\Rightarrow 2^a = 2k = \log n \Rightarrow a = \log(\log n)$

Therefore, number of elementary operations $= C \cdot \left(\frac{\log n(\log n+1)}{2} + \sqrt{n} + (n - \sqrt{n}) \cdot n + \log(\log n)\right)$