Part1 – 1

* Inner loop:

$$\sum_{k=0}^{j+5} 1 = j + 5 - 0 + 1 = j + 6$$

* Outer loop:

$$\sum_{j=1}^{n} (j+6) = \sum_{j=1}^{n} j + \sum_{j=1}^{n} 6 = \frac{n(n+1)}{2} + 6n = \frac{n^2}{2} + \frac{n}{2} + \frac{12n}{2} = \frac{n^2}{2} + \frac{13n}{2}$$

* The answer :

$$\frac{n^2}{2} + \frac{13n}{2}$$

Part1 – 2

* Inner loop:

$$\sum_{j=i+1}^{n} 2 = 2 \sum_{j=i+1}^{n} 1 = 2(n - (i+1) + 1) = 2(n - i)$$

* Outer loop:

$$\sum_{i=0}^{n-1} (1 + 2(n-i)) = \sum_{i=0}^{n-1} 1 + \sum_{i=0}^{n-1} 2n - \sum_{i=0}^{n-1} 2i = n + 2n^2 - (n^2 - n) = n^2 + 2n$$

$$\sum_{i=0}^{n-1} 1 = n - 1 - 0 + 1 = n$$

$$\sum_{i=0}^{n-1} 2n = 2n \sum_{i=0}^{n-1} 1 = 2n^2$$

$$\sum_{i=0}^{n-1} 2i = 2 \sum_{i=0}^{n-1} i = 2\frac{(n-1)n}{2} = n^2 - n$$

* The answer :

$$n^2 + 2n$$

Part1 – 3

* Inner loop:

$$\sum_{j=0}^{i^2-1} 1 = i^2 - 1 - 0 + 1 = i^2$$

* Outer loop:

$$1 + \sum_{i=1}^{n}(1 + i^2) = 1 + \sum_{i=1}^{n} 1 + \sum_{i=1}^{n} i^2 = 1 + n + \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{n^3}{3} + \frac{n^2}{2} + \frac{7n}{6} + 1$$

* The answer :

$$\frac{n^3}{3} + \frac{n^2}{2} + \frac{7n}{6} + 1$$

Part2 – 4

$Does\ 5 \cdot (n+2)^2 = \Omega(n \log n)$?

$f(n) = 5 \cdot (n+2)^2$

$g(n) = n \log n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{5 \cdot (n+2)^2}{n \log n}$$

Apply l'Hôpital's rule

$$\lim_{n\to\infty} \frac{f'(n)}{g'(n)} = \lim_{n\to\infty} \frac{10(n+2)}{\log n + n \frac{1}{n \ln 10}}$$

Apply one more l'Hôpital's rule

$$\lim_{n\to\infty} \frac{10}{\frac{1}{n \ln 10}} = \lim_{n\to\infty} 10 \cdot n \ln 10 = \infty$$

$$if\ \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty\ then\ f(n) = \Omega\big(g(n)\big)$$

$\therefore 5 \cdot (n+2)^2 = \Omega(n \log n)$

* The answer is TRUE

Part2 – 5

$Does\ 8^{\log_2 n + \log_2 \log_2 n} = \Omega(n^3)$?

$f(n) = 8^{\log_2 n + \log_2 \log_2 n}$

$f(n) = 2^{3^{\log_2 n}} \cdot 2^{3^{\log_2 \log_2 n}} = 2^{\log_2 n^3} \cdot 2^{\log_2 (\log_2 n)^3} = n^3 \cdot (\log_2 n)^3$

$g(n) = n^3$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^3 \cdot (\log_2 n)^3}{n^3} = \lim_{n\to\infty} (\log_2 n)^3 \ = \infty$$

$$if\ \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty\ then\ f(n) = \Omega\big(g(n)\big)$$

$\therefore 8^{\log_2 n + \log_2 \log_2 n} = \Omega(n^3)$

* The answer is TRUE

Part2 – 6

$Does\ n\log_{16}n = O(n\ln n)$?

$f(n) = n\log_{16}n$

$g(n) = n\ln n$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n\log_{16}n}{n\ln n} = \lim_{n\to\infty}\frac{\log_{16}n}{\ln n}$$

Apply l'Hôpital's rule

$$\lim_{n\to\infty}\frac{\frac{1}{n}\cdot\frac{1}{\ln16}}{\frac{1}{n}} = \lim_{n\to\infty}\frac{1}{\ln16} = k > 0$$

$$if\ \lim_{n\to\infty}\frac{f(n)}{g(n)} = k, k > 0\ \ then\ f(n) = \Theta(g(n))$$

$\therefore n\log_{16}n = \Theta(n\ln n)$

* The answer is FALSE

Part2 – 7

$Does\ (3n^2 - 10n) = \Theta(n^2)?$

$f(n) = 3n^2 - 10n$

$g(n) = n^2$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{3n^2 - 10n}{n^2} = \lim_{n\to\infty} \frac{3 - \frac{10}{n}}{1} = \lim_{n\to\infty} 3 - \frac{10}{n} = 3 > 0$$

$$if \lim_{n\to\infty} \frac{f(n)}{g(n)} = k, k > 0\ \ then\ f(n) = \Theta(g(n))$$

$\therefore 3n^2 - 10n = \Theta(n^2)$

* The answer is TRUE

Part2 – 8

$Does\ \ n\log n = \Omega\left(n^{\frac{11}{8}}\right)$?

$f(n) = n\log n$

$g(n) = n^{\frac{11}{8}}$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n\log n}{n^{\frac{11}{8}}} = \lim_{n\to\infty}\frac{\log n}{n^{\frac{3}{8}}}$$

Apply the l'Hôpital's rule

$$\lim_{n\to\infty}\frac{\frac{1}{n\ln 10}}{\frac{3}{8}n^{-\frac{5}{8}}} = \lim_{n\to\infty}\frac{n^{-\frac{3}{8}}\cdot\frac{1}{\ln 10}}{\frac{3}{8}} = \lim_{n\to\infty}\frac{\frac{1}{\ln 10}}{\frac{3}{8}\cdot n^{\frac{3}{8}}} = \frac{8}{3\cdot\ln 10\cdot n^{\frac{3}{8}}} = 0$$

$if\ \lim_{n\to\infty}\frac{f(n)}{g(n)} = 0\ \ then\, f(n) = O(g(n))$

$\therefore n\log n = O\left(n^{\frac{11}{8}}\right)$

* The answer is FALSE

Part2 – 9

Q. Let f(n) and g(n) be non-negative functions of n. Prove using the definitions of O, **Ω**, and Θ — not using limit tests — that if g(n) = **Ω**(f(n)), then f(n) + g(n) = Θ(g(n))

$f(n) \geq 0$

$g(n) \geq 0$

$(i)\ \ f(n) + g(n) \geq g(n) \because f(n) \geq 0$

(ii)

$if\ \ g(n) = \Omega\big(f(n)\big)\ then\ f(n) = O(g(n))$

$f(n) \leq c \cdot g(n), c > 0, n_0 > 0, all\ n \geq n_0$

$f(n) + g(n) \leq c \cdot g(n) + g(n)$

$f(n) + g(n) \leq (c+1) \cdot g(n)$

$(i)(ii)\ 1 \cdot g(n) \leq f(n) + g(n) \leq (c+1)g(n)$

$\therefore f(n) + g(n) = \Theta(g(n))$

Part2 – 10

Q. Let f(n) and g(n) be non-negative functions of n. If f(n) = Θ(g(n)), does f(n)/g(n) = Θ(1)? Justify your answer.

$f(n) \geq 0$

$g(n) \geq 0$

$if\ f(n) = \Theta\big(g(n)\big)\ then\ c_1 g(n) \leq f(n) \leq c_2 g(n), c_1 > 0, c_2 > 0, n_0 > 0, for\ all\ n \geq n_0$

Divide above equation by $g(n)$

$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

$$c_1 \cdot 1 \leq \frac{f(n)}{g(n)} \leq c_2 \cdot 1$$

$$\therefore \frac{f(n)}{g(n)} = \Theta(1)$$

Part3 – 11

$A = 4 \cdot 10^{-6} n^2 \ seconds$

$B = 10^{-4} n \log_2 n \ \ seconds$

We know that the time complexity as follows

$$A = O(n^2)$$

$$B = O(n \log_2 n)$$

If n becomes bigger algorithm B is much more efficient compared to A. But there is some point where algorithm A is faster than B.

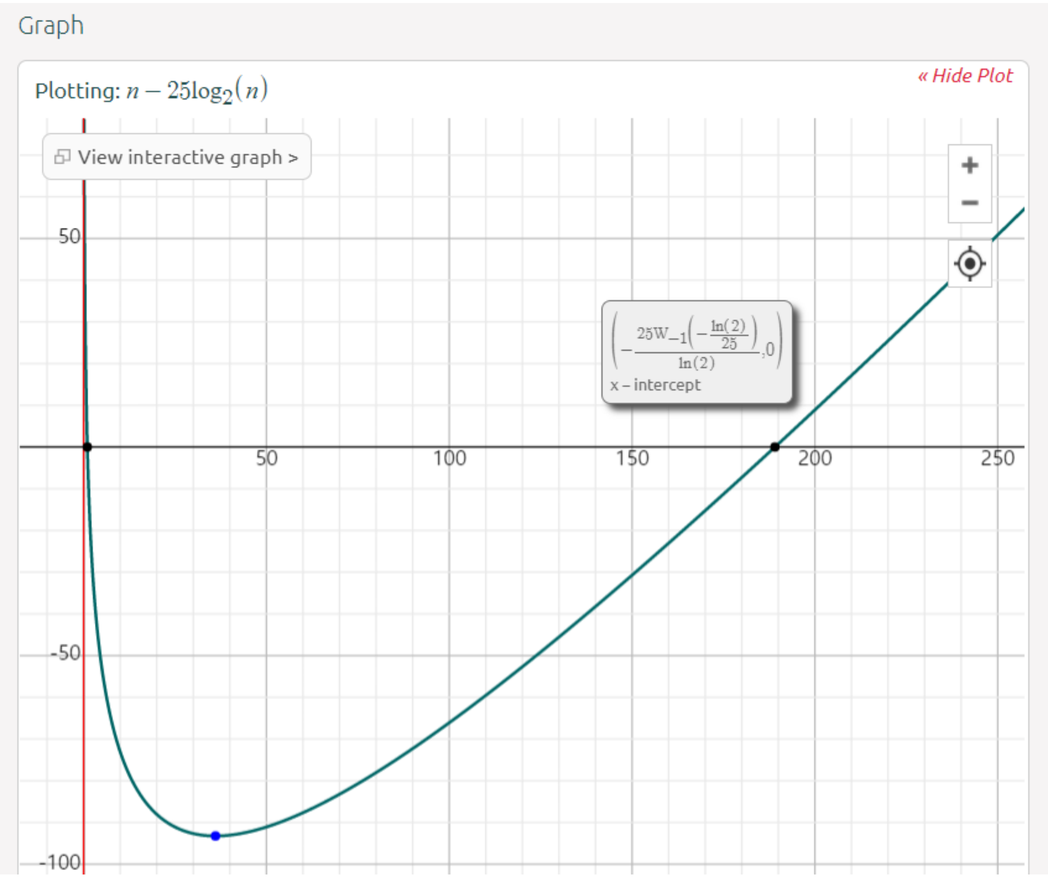
If we figure out $A = B$ and get a value of n_0 where algorithm B is strictly faster than algorithm A for all $n \geq n_0$.

$4 \cdot 10^{-6} \cdot n^2 = 10^{-4} n \log_2 n$

$n = 25 \log_2 n$

$n - 25 \log_2 n = 0$

$n_0 \ is \ shown \ below$



Part3 – 12

$B = 10^{-4} n \log_2 n \ seconds$

$C = 1.4 \cdot 10^{-4} n \ seconds$

We know that the time complexity as follows

$$B = O(n \log_2 n)$$

$$C = O(n)$$

If n becomes bigger algorithm C is much more efficient compared to B. But there is some point where algorithm B is faster than C.

If we figure out $B = C$ and get a value of n_1 where algorithm C is strictly faster than algorithm B for all $n \geq n_0$.

$10^{-4} n \log_2 n = 1.4 \cdot 10^{-4} n$

$\log_2 n = 1.4$

$n_1 = 2^{1.4}$

Part3 – 13

As a result of Question 11, we know when algorithm A is faster than B when $n < n_0$.

As a result of Question 12, we know when algorithm B is faster than C when $n < n_1$.

When the input phone number is less than n_0 and then use algorithm A.

When the input phone number is more than n_0 and less than n_1 and then use algorithm B.

When the input phone number is more than n_1 and then use algorithm C.

Part3 – 14

We need to find out how many seconds it will take when using algorithm A.

$Let's\ put\ 10^8\ in\ the\ equation$

$A\ Algorithm = 4 \cdot 10^{-6} n^2 = 4 \cdot 10^{-6} (10^8)^2 = 40{,}000{,}000{,}000\ seconds$

So, if there is one processor and using algorithm A and then it will take $40{,}000{,}000{,}000\ seconds.$

We need to do it in an hour, so divide it by 3600 seconds.

We get $\frac{40{,}000{,}000{,}000}{3600} \approx 11{,}111{,}111.111111$

As a result, we need more than 11,111,112 processors to do it in an hour with algorithm A.

We need to find out how many seconds it will take when using algorithm B.

$Let's\ put\ 10^8\ in\ the\ equation$

$B\ Algorithm = 10^{-4} n \log_2 n = 10^{-4}(10^8) \log_2(10^8) = 10^4 \log_2(10^8) = 80000 \log_2 10$

$\approx 80000 \cdot 3.321928 \approx 265754.24\ seconds$

$\because \log_2 10 \approx 3.321928$

So, if there is one processor and using algorithm B and then it will take about $265754.24\ seconds.$

We need to do it in an hour, so divide it by 3600 seconds.

We get $\frac{265754.24}{3600} \approx 73.82$

As a result, we need more than 74 processors to do it in an hour with algorithm B.

We need to find out how many seconds it will take when using algorithm C.

$Let's\ put\ 10^8\ in\ the\ equation$

$C\ Algorithm = 1.4 \cdot 10^{-4} n = 1.4 \cdot 10^{-4} \cdot 10^8 = 1.4 \cdot 10000 = 14000\ seconds$

So, if there is one processor and using algorithm C and then it will take about $14000\ seconds.$

We need to do it in an hour, so divide it by 3600 seconds.

We get $\frac{14000}{3600} \approx 3.88$

As a result, we need more than 4 processors to do it in an hour with algorithm C.