Q4

a) $n \log n^3$

$3 n \log n$

for $c = 1, n_0 \geq 0$ for $c = 4, n_0 > 0$

$\underset{f(x)}{n \log(n)} \leq \underset{g(x)}{3 n \log n}$ $g(x) \leq f(x)$

$\therefore$ True

b) $f(n) = \frac{1}{3} n^2 + 10n - 2$ $g(n) = n^2$

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

$c \cdot n^2 \leq \frac{1}{3} n^2 + 10n - 2$

$c \cdot \frac{2}{3} n^2 \leq 10n - 2$

$c \cdot 2n^2 \leq 30n - 6$

$c \cdot n^2 \leq 15n - 3$

~~for any value of c or n_0~~

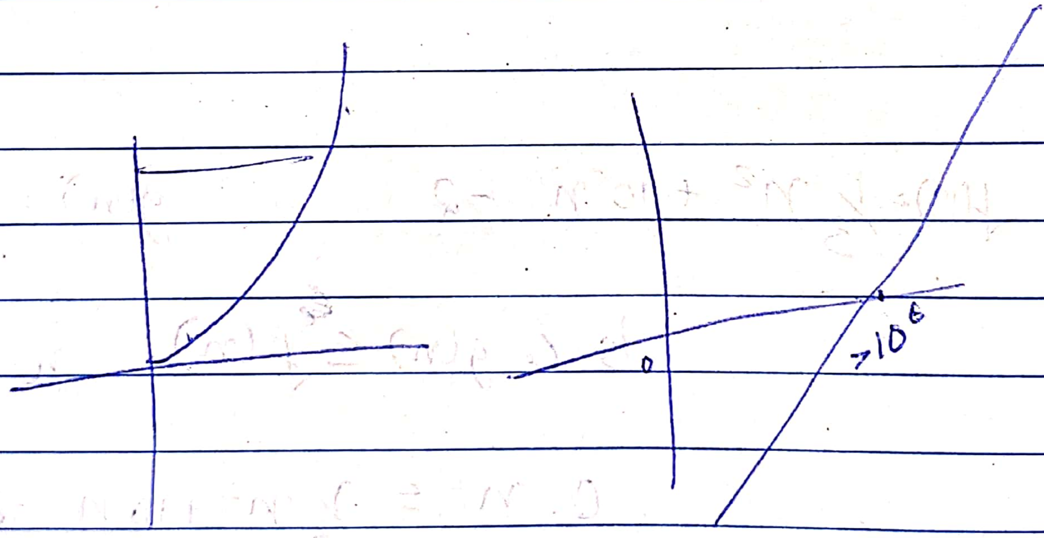
if we take c ~~as~~ n_0 as any value, it will not always hold true.

c. $f(n) = n^3 \quad g(n) = 100n^2 + 10^6 n$

$$f(n) \le c \cdot g(n)$$

$$n^3 \le c \cdot 100n^2 + 10^6 n$$

$$n^2 \le c \cdot 100n + 10^6$$



So for any combination of c & n_0, n^2 will always be greater because it grows exponentially.

d. $f(n) = \sqrt{n^2+1}$ $\quad g(n) = n^2+3$

$$c \cdot g(n) > f(n)$$

$$c \cdot n^2 + 3 > \sqrt{n^2+1} \quad \text{— (1)}$$

Let $c = 100$ & $n_0 = 1$

$$100 \cdot 1 + 3 > \sqrt{1+1}$$

$$103 > \sqrt{2}$$

$\therefore$ for all values of ~~n_0~~ $\forall\, n > n_0$, the equation (1) holds true.

e. $f(n) = n^2 \sin(1/n) + 2n$

$g(n) = n^2$

$$f(n) > g(n) \cdot c$$

$$n^2 \sin(1/n) + 2n > n^2 \cdot c$$

range of sin is $0 \to 1$

when sin = 1

$n = 1/90, 1/180, 1/270, 1/360 \ldots$

$= \frac{2n}{\pi}$

$n^2 + 2n \geq n^2$

For all cases, this won't hold true.
Therefore it is false.

Q5.

a. The inner while loop here executes only once always. Because, let's take $n = 50$. $i = 0$ first $j = 0+1$. Now for loop will move to 2nd iteration. $i = 1$, $j = 1+1$. And so on. The for loop will go through n iterations but while only once always. Therefore the complexity is $O(n)$.

b. Here the essential operations are all $O(1)$. Always n is only being subtracted by 5. So it is $O(\frac{n}{5})$ which is $O(n)$

c. Here it is very straightforward. The ~~it~~ ~~co~~ i value is being updated by ~~x~~ 2 times. Therefore ~~&~~ it will run only ~~log base 2~~ $\log_2 n$ no. of times.

d. Here the for loop will run n^2 times ~~by~~ the but similar to the last question we are dividing instead of multiplying with 2. Therefore it will be $\log n$ times inside the loop

The time complexity will be $n^2 \log n$.

e. ~~def~~ $n = 512$

i — 4 executions

2
— 1
4
— 2
16
— 3
256
— 4
256×256

$n = 64$ — 3 executions

i
2
— 1
4
— 2
16
— 3
256

$n = 16$ — 2 executions

i
2
— 1
4
— 2
16

The no. of executions will always follow $\log_2 n$ or $\log (n)^{1/2}$ or $\log \sqrt{n}$