Q.1 if $f(n) = O(g(n))$

① then $\log_2 f(n) = O(\log_2 g(n))$

assuming $n \in N$ with $n \geq N$, $c \in R$

$$0 \leq f(n) \leq c g(n)$$

But $\log_2$ is order-preserving

$$\log_2 f(n) \leq \log_2 c\, g(n)$$
$$= \log_2 C + \log_2 g(n)$$

$\because d \in R > 0$, $\log_2 f(n) \leq d \log_2 (g(n))$

$$\log_2 C + \log_2 g(n) \leq d \log_2 g(n)$$

$$\frac{\log_2 C}{\log_2 g(n)} + 1 \leq d$$

but $\log g(n) \to 0$ when $n \to \infty$

$\therefore 2\left(1 + \frac{1}{n}\right) \in O\left(1 + \frac{1}{n}\right)$

but $\log_2 2 + \log_2\left(1 + \frac{1}{n}\right) \notin O\left(\log_2\left(1 + \frac{1}{n}\right)\right)$

So proof does not exist

(2) if $f(n) = O(g(n))$ then $3^{f(n)} = O(3^{g(n)})$

$f(n) = 3(\log_3 n)$ & $g(n) = \log_3 n$ — (1)

$f(n) = O(g(n))$ — (1)

$\therefore 3\log_3 n \leq c(\log_3 n)$

now $3^{f(n)} = 3^{3\log_3 n} = 3^{\log_3 n^3} = n^3$

$3^{g(n)} = 3^{\log_3 n} = n \quad n^3 \neq n$

So proof does not exists.

(3) $f(n)^3 = O(g(n)^3)$

Let $f(n) = n \qquad g(n) = n^2$

$\therefore n^3 = O((n^2)^3)$

$\therefore n^3 = O(n^6)$ is true

$\therefore$ So proof exists.

Q.4 The correct order is

$$\boxed{f_2 < f_3 < f_1 < f_4 < f_5}$$

The easy one to tell are f_1, f_2, f_3 polynomial functions.

$$(2n)^{1.6} < n^{4.5} + 87 < n^{4.6}$$

In f_1 & f_4

$$n^{4.6} < 40^n \quad \text{when } n \to \infty$$

and now for f_4 & f_5 it is easy $40^n < 120^n$.

Q.2

$A_1 \to 10^{-4} \times 2^n$ s to solve n

$A_2 \to 10^{-2} \times n^3$ s to solve n

~~(i) Seconds in a year = 365 × 24 × 60 × 60 Seconds~~

~~$n \longrightarrow 10^{-4} \times 2^n \quad 10^{-2} \times n^3$~~

~~$x \longrightarrow 365 \times 24 \times 60 \times 60$~~

~~$x = \dfrac{365 \times 24 \times 60 \times 60 \times n}{10^{-4} \times 2^n \quad 10^{-2} \times n^3}$~~

(i) $365 \times 24 \times 60 \times 60 = A_2$

$365 \times 24 \times 60 \times 60 = 10^{-2} \times n^3$

$n^3 = 365 \times 24 \times 60 \times 60 \times 100$

$n \approx 1466.45$

$$\boxed{n \approx 1466}$$

5.87

(ii) Now A2 is running on 100 times faster machine

$\therefore$ it will take $\frac{10^{-2} \times n^3}{100}$ seconds for 'n'.

$\therefore$ $10^{-4} \times n^3$ seconds for n on M2

$\therefore$ $365 \times 24 \times 60 \times 60 = 10^{-4} \times n^3$

$\boxed{n \approx 6807}$

(iii) for $n < 20$

$A_1 = 10^{-4} \times 2^n$ $A_2 = 10^{-2} \times n^3$

[A1 & A2 intersects at point 19.5
So question should be below 19.5 not 20]

[But instance can't be decimal]
[So que is correct]

	A1	A2
for n 1	2×10^{-4}	10^{-2}
n 2	4×10^{-4}	8008×10^{-4}
n 3	8×10^{-4}	2700×10^{-4}

$\therefore$ A1 will produce result faster than A2

Q. 3

(i) $3n^3 + 1000 = O(n^2)$

$f(n)$ is $O(g(n))$ if there exist constant $c > 0$ and $n_0 \geq 0$ so that for all $n \geq n_0$

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

But here $3n^3 + 1000 \geq c \cdot n^2$

which is violating upper bond defination state

$\therefore$ So proof does not exist.

(ii) $2n^2 \log(n) = \Theta(n^2)$ if $\begin{cases} 2n^2 \log(n) = O(n^2) & — ① \\ 2n^2 \log(n) = \Omega(n^2) & — ② \end{cases}$

$n^2 = k$ $\therefore 2k \log(k^{1/2}) = O(k)$ — Solving ①

$\therefore k \log(k) \neq O(k)$ — not true

$\therefore$ Proof does not exist.

(iii) $3^n n^4 + 8 * 4^n n^3 = O(3^n n^4)$

Let's assume $3^n n^4$ is tightly bounding function.

$\therefore 3^n n^4 + 8 * 4^n n^3 \leq c \cdot 3^n n^4$

$\therefore 8 \times 4^n n^3 \leq c \cdot 3^n n^4$

$8 \times 4^n \leq c \cdot 3^n \cdot n$

But $4^n \geq 3^n$

$\therefore$ Proof does not exist