Q1: Prove that: **$n^3 = O(100n^3)$**

We want to show that there exist **constants $c > 0$** and **$n_0 \geq 0$** such that, for all $n \geq n_0$:

$n^3 \leq c * 100n^3$

Let's choose:

$c = 0.5$
$n_0 = 1$

Now check the inequality for all $n \geq n_0$:

$n^3 \leq 0.5 * 100n^3$
$n^3 \leq 50n^3$

This is clearly true for all $n \geq 1$.

Therefore:

By the definition of Big-O, we conclude:

$n^3 = O(100n^3)$

Q2: Prove that

$n^2 * n^3 \neq O(n^4)$

Let:

$f(n) = n^2 * n^3 = n^5$
$g(n) = n^4$

We want to check if:

$f(n) \leq c * g(n)$ for some $c > 0$ and sufficiently large n

In other words:

$n^5 \leq c * n^4$

But this simplifies to:

$n \leq c$

Which is **false** as $n \to \infty$. No constant c can satisfy this forever.

More formally, take the limit:

$\lim (n \to \infty) [f(n) / g(n)] = \lim (n \to \infty) [n^5 / n^4] = \lim (n \to \infty) n = \infty$

Since the limit diverges to infinity, it **violates** the Big-O definition.

Therefore:

There does **not** exist any constant c such that:

$n^5 \leq c * n^4$ for all large n

So we conclude:

$n^2 * n^3 \neq O(n^4)$

Q3: Prove that

$n^2 \cdot 2^n = O(2^{2n})$

Let:

$f(n) = n^2 * 2^n$
$g(n) = 2^{2n} = (2^n)^2 = 4^n$

We want to find constants **$c > 0$** and **$n_0 \geq 0$** such that:

$n^2 * 2^n \leq c * 4^n$ for all $n \geq n_0$

Note that:

$4^n = 2^{2n}$
So: $n^2 * 2^n \leq c * 2^{2n}$

Divide both sides by 2^n:

$n^2 \leq c * 2^n$

Now, does this inequality hold for large n?

Yes — because **2^n grows faster than any polynomial**, including n^2.

Then, let:

$c = 1$
$n_0 = 10$

Now test:

$n^2 \leq 2^n$
$\rightarrow 10^2 = 100 \leq 2^{10} = 1024$ / (checks out)
$\rightarrow 20^2 = 400 \leq 2^{20} = 1{,}048{,}576$ / (checks out)

So the inequality:

$n^2 * 2^n \leq c * 4^n$

holds for $n \geq 10$.

Therefore:

By the definition of Big-O, we conclude:

$n^2 * 2^n = O(2^{2n})$

Q4: Disprove that

2^(2n) = O(3^n)
(i.e., prove that 2^(2n) grows faster than 3^n)

Let:

f(n) = 2^(2n) = (2^2)^n = 4^n
g(n) = 3^n

We want to test if:

f(n) ≤ c * g(n)   for some constant c > 0 and all $n \geq n_0$
→ 4^n ≤ c * 3^n

From here;

lim (n → ∞) [f(n) / g(n)] = lim (n → ∞) (4^n / 3^n) = lim (n → ∞) (4/3)^n = ∞

Since the limit **diverges to infinity**, that means:

- f(n) grows strictly faster than any constant multiple of g(n)
- The inequality f(n) ≤ c * g(n) eventually **fails** for any c

Therefore:

There does **not** exist any constant c such that:

2^(2n) ≤ c * 3^n   for all large n

So:

2^(2n) ≠ O(3^n)

Q5: Prove that

n * log(n^5) = O(n * log(n))

Let:

f(n) = n * log(n^5)
g(n) = n * log(n)

Using the logarithm identity:

log(n^5) = 5 * log(n)

So:

f(n) = n * log(n^5) = n * 5 * log(n) = 5 * n * log(n)

Thus:

f(n) = 5 * g(n)

Which gives:

f(n) ≤ 5 * g(n)   for all n ≥ 1

Let:

c = 5
n_0 = 1

Then:

f(n) ≤ c * g(n)   for all n ≥ n_0

Therefore:

By the definition of Big-O:

n * log(n^5) = O(n * log(n))