# ASYMPTOTIC NOTATION

**1<log n < (root(n) < n <nlogn <n^2 <n^3 …………………….. 2^n <3^n <…..< n^n**

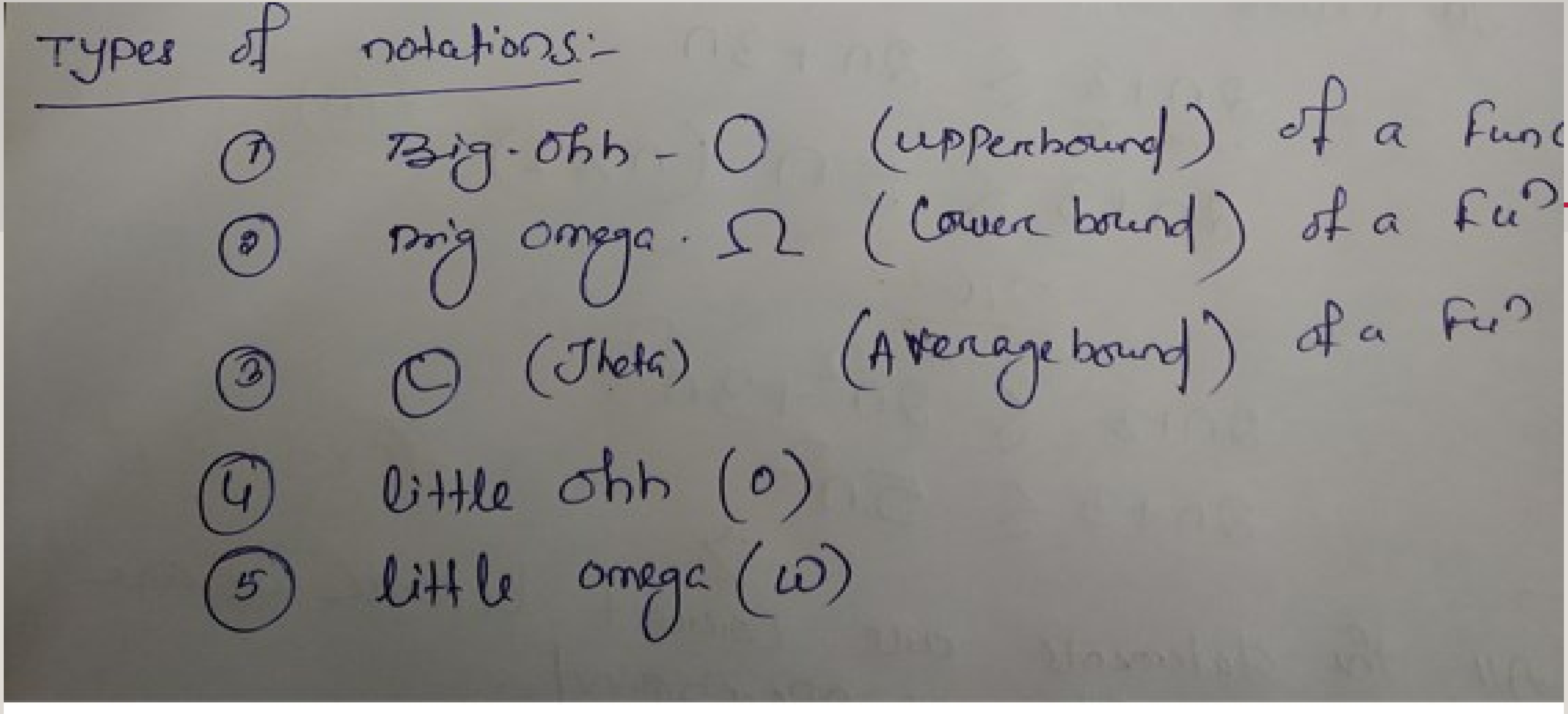
| | |
|---|---|
| Smaller | Greater |

- The notations are used for representing the symbol form of a function or showing the class of a function.

- Asymptotic notations are the mathematical notations used to describe running time of an algorithm when the input tends towards a particular value or limiting value.

- Ex: When the input array is in sorted order, the time taken by the algorithm is linear.

- When the input array is in reverse order, it takes maximum time.



Types of notations:-

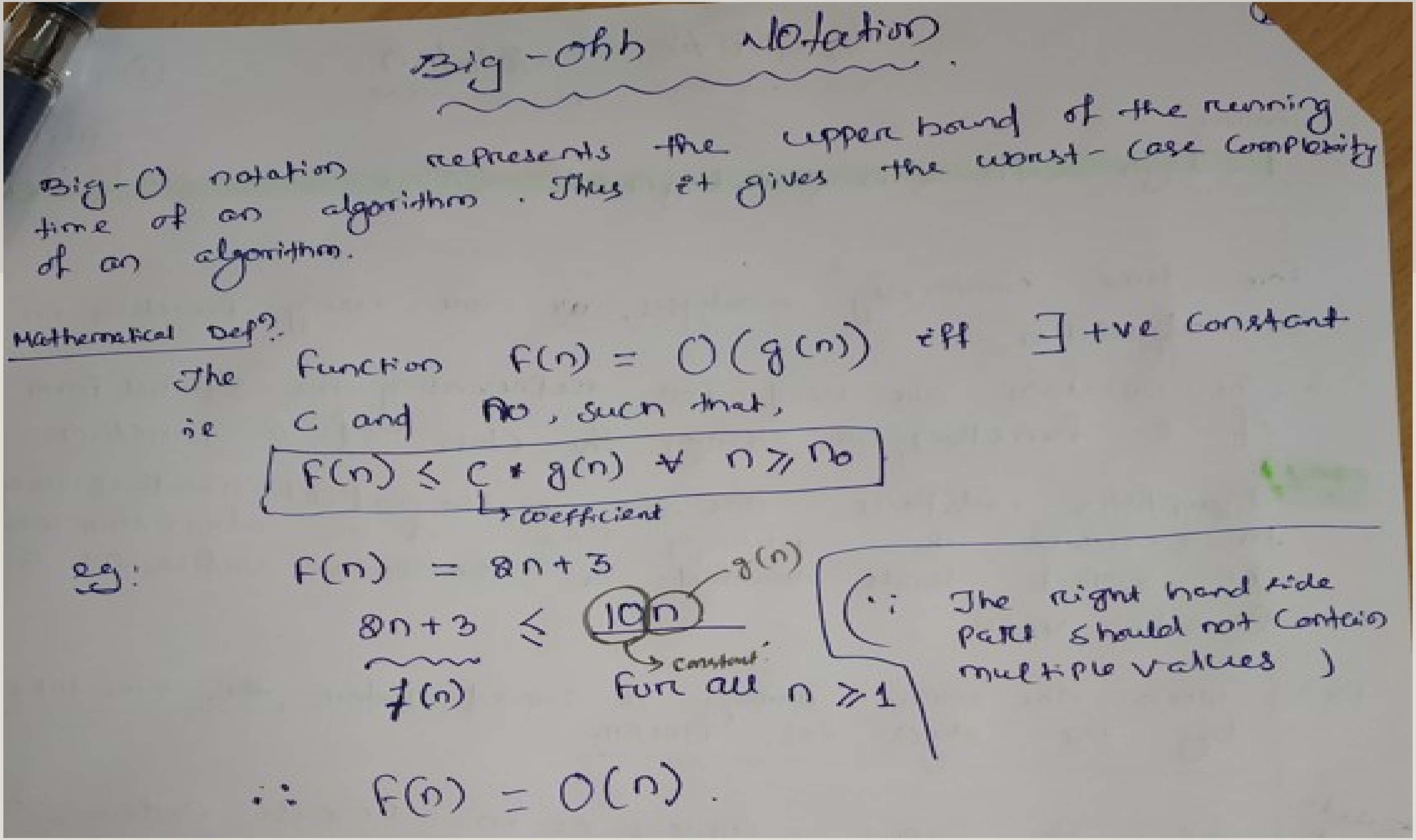
① Big-Ohh - O (upperbound) of a func

② Big omega · Ω (Lower bound) of a fu$^{n}$

③ Θ (Theta) (Average bound) of a fu$^{n}$

④ little ohh (o)

⑤ little omega (ω)



# Big-ohh Notation.

Big-O notation represents the upper band of the running time of an algorithm. Thus it gives the worst-case complexity of an algorithm.

Mathematical Def$^n$:-

The function $f(n) = O(g(n))$ iff $\exists$ +ve constant

ie c and $n_0$, such that,

$$\boxed{f(n) \le \underbrace{c}_{\text{coefficient}} * g(n) \ \forall \ n \ge n_0}$$

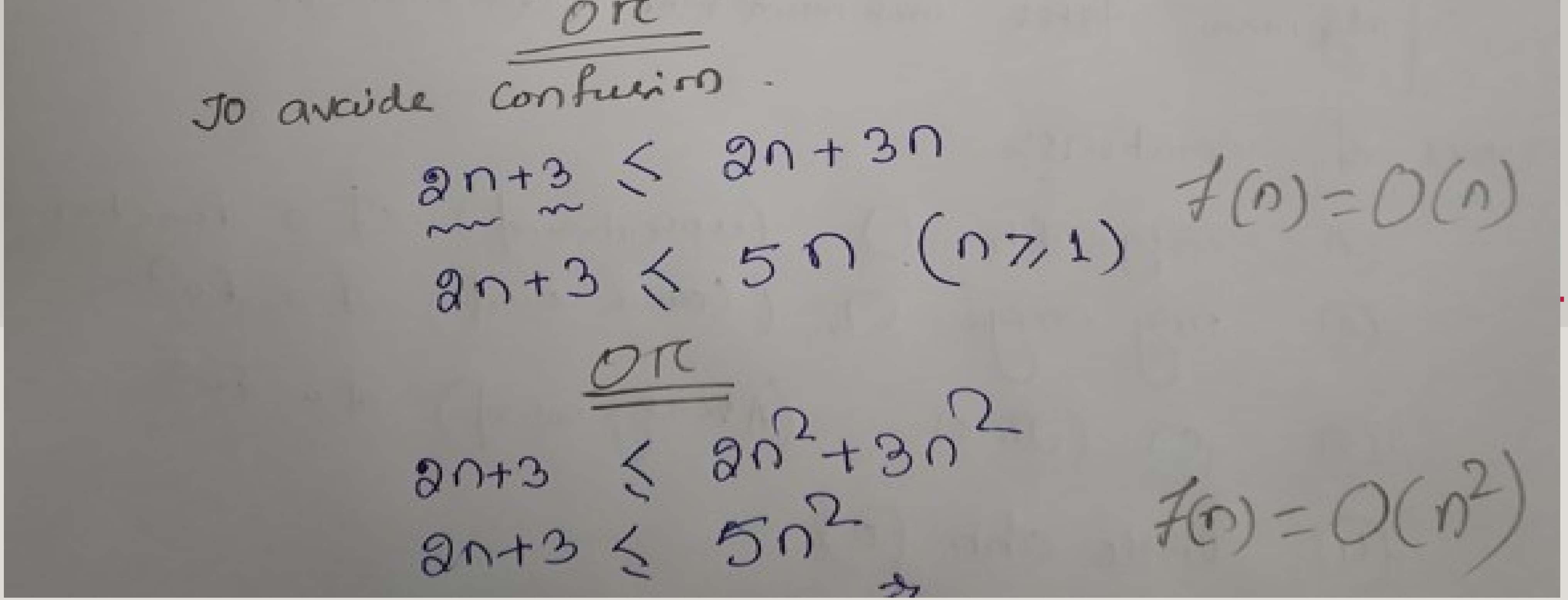
eg: $f(n) = 8n+3$

$$\underbrace{8n+3}_{f(n)} \le \overbrace{\underbrace{10}_{\text{constant}}\, n}^{g(n)}$$

for all $n \ge 1$

( ∵ The right hand side part should not contain multiple values )

$\therefore \ f(n) = O(n)$.



or

To avoide confusion .

$2n+3 \leq 2n+3n$

$2n+3 \leq 5n \quad (n \geq 1)$ $\quad f(n)=O(n)$

or

$2n+3 \leq 2n^2+3n^2$

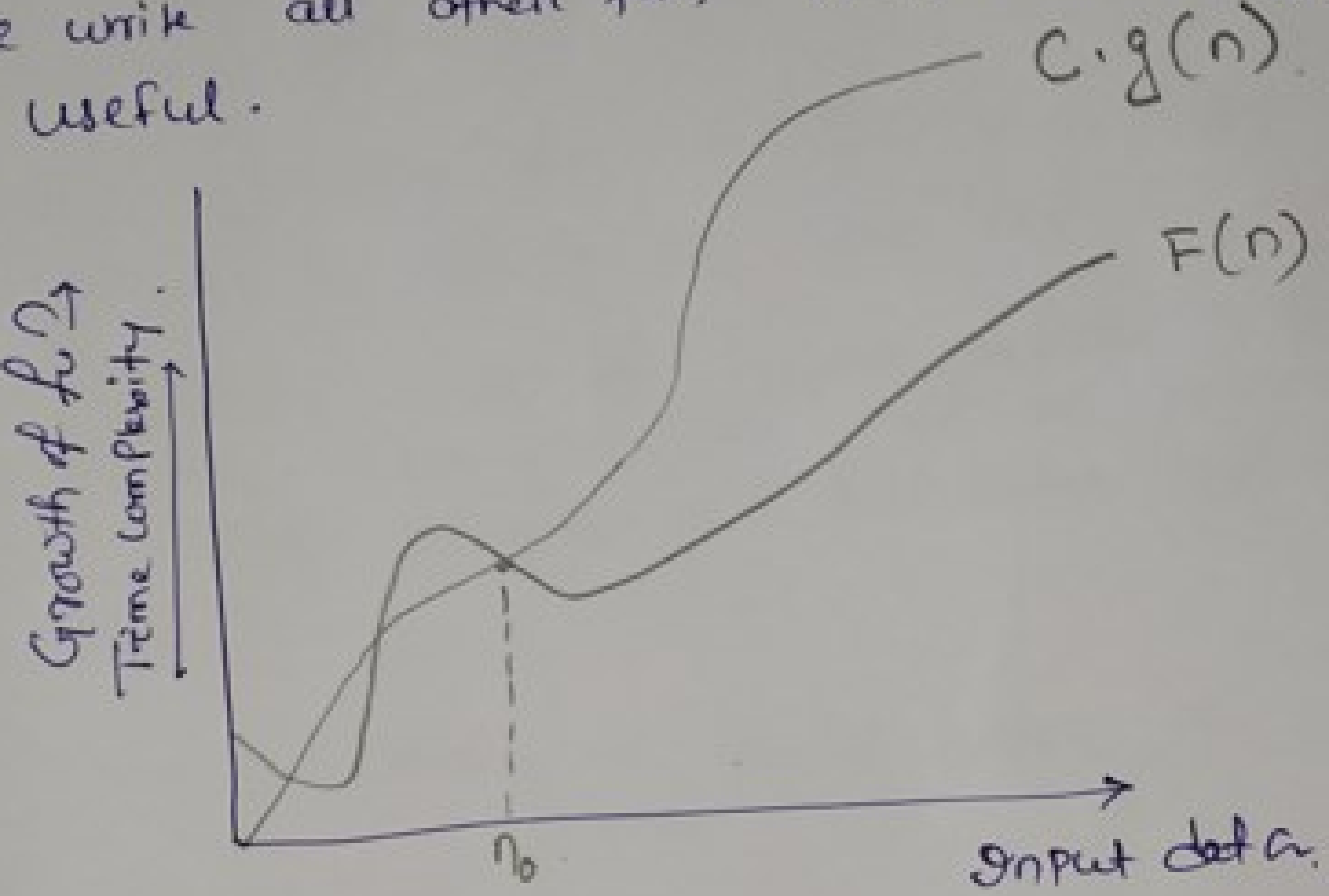
$2n+3 \leq 5n^2$ $\quad f(n)=O(n^2)$

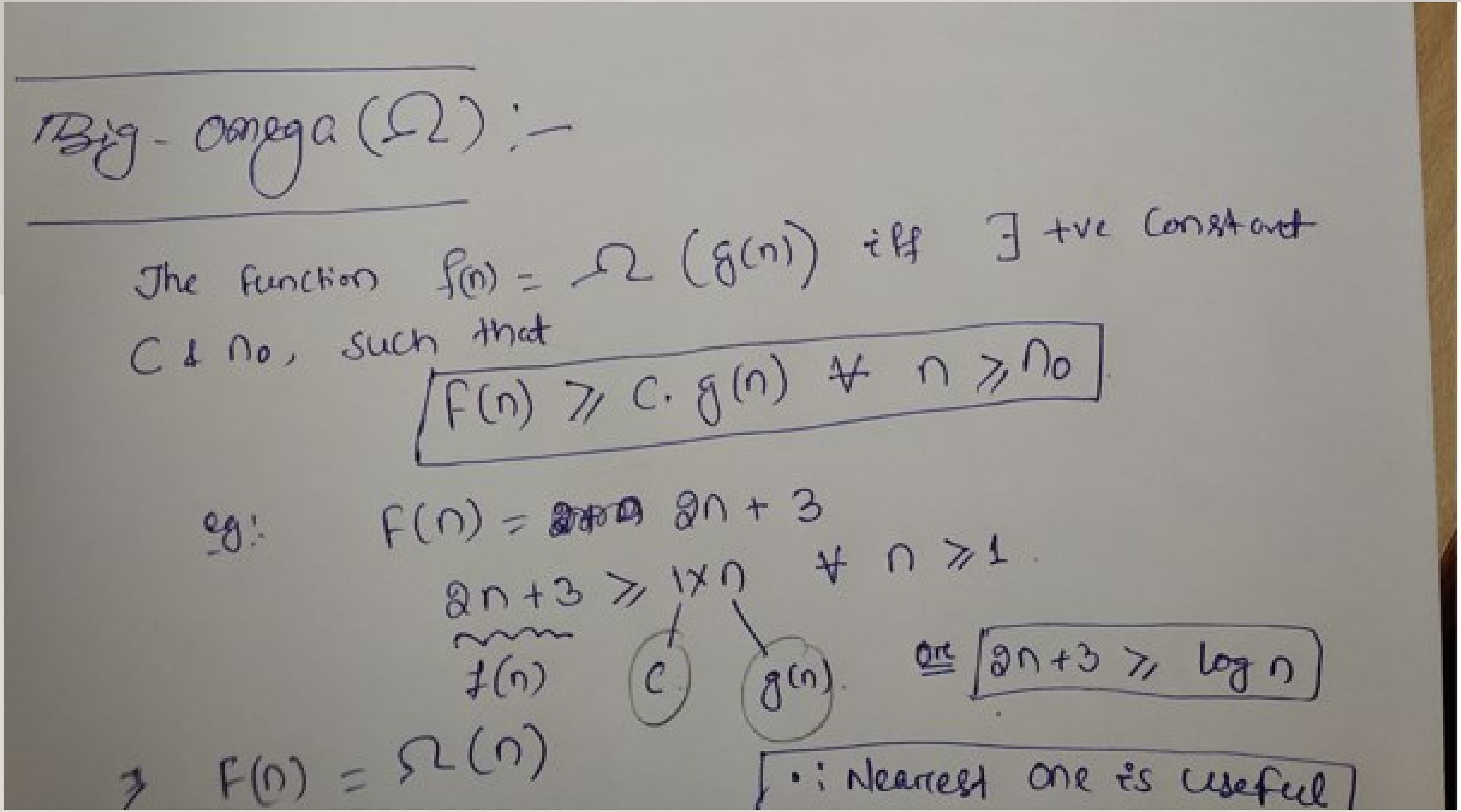
All the above methods are correct, because we are looking for upper bound conditions.

Q). If all the statements are correct, then which one is appropriate.

Sol: Take the closest function always for analysis.

**NOTE** :- If we write all other fn. it is not wrong, but it is not useful.





## Big - omega (Ω) :-

The function $f(n) = \Omega(g(n))$ iff $\exists$ +ve constant $c$ & $n_0$, such that

$$\boxed{F(n) \geq c \cdot g(n) \ \forall \ n \geq n_0}$$

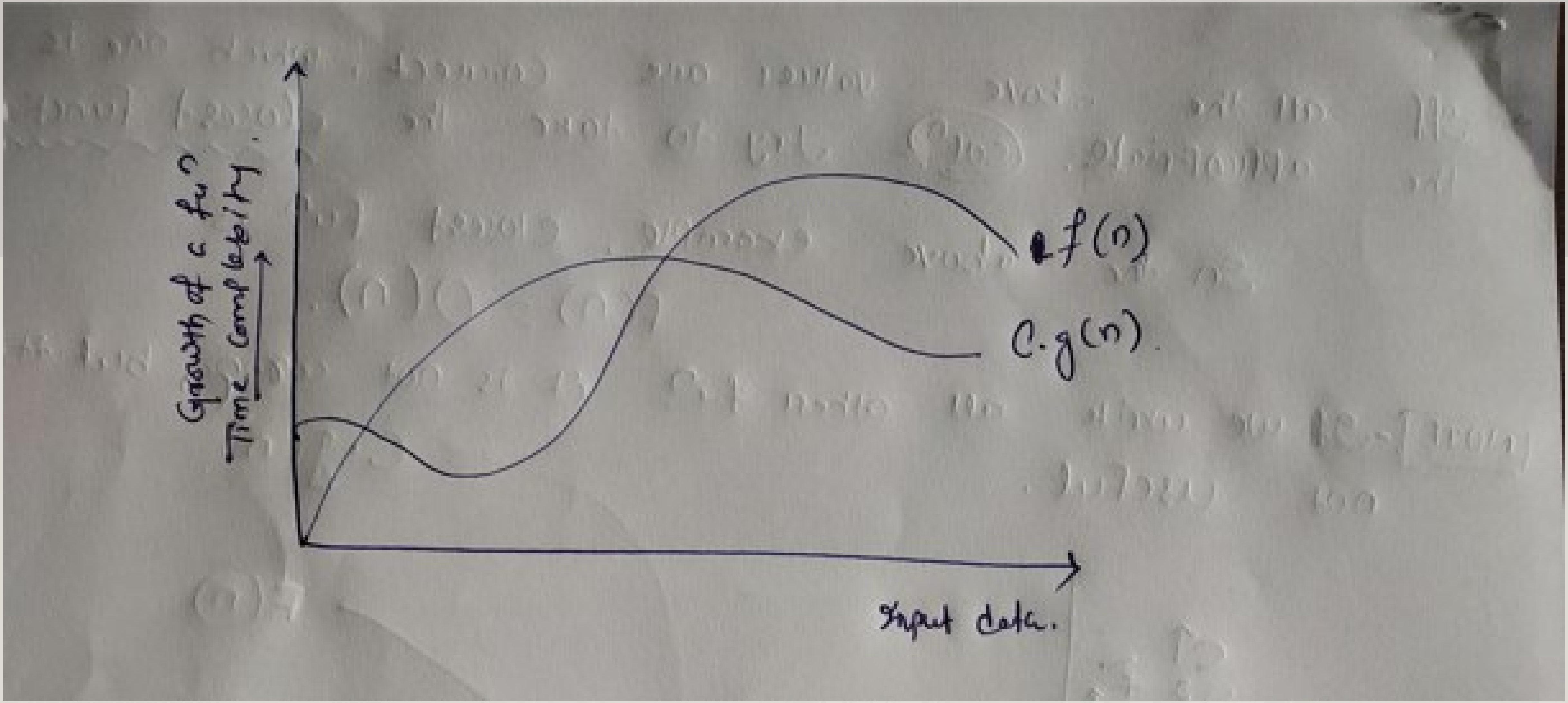
eg: $F(n) = 2n + 3$

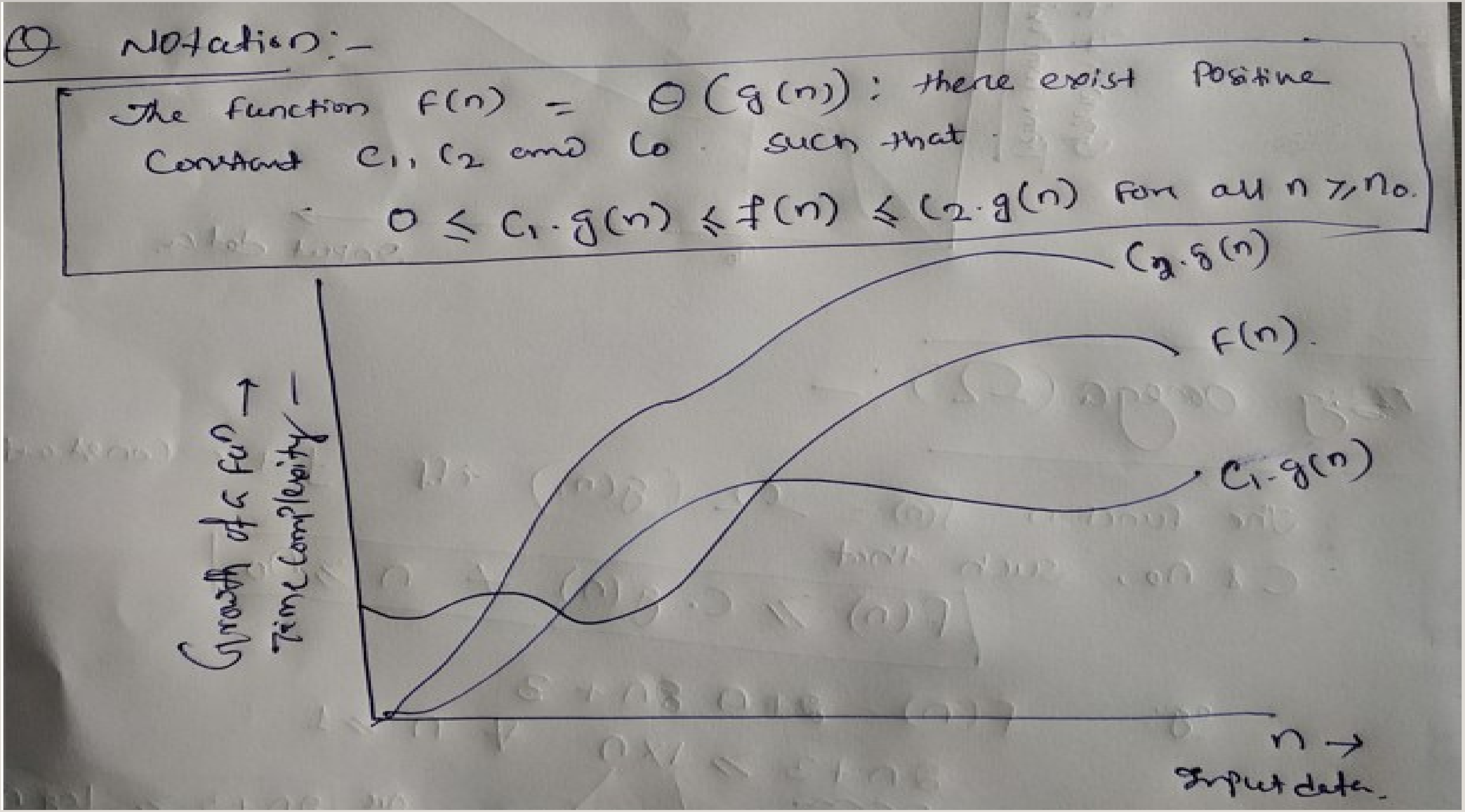
$$\underbrace{2n+3}_{f(n)} \geq \underset{c}{1} \times \underset{g(n)}{n} \quad \forall \ n \geq 1.$$

or $\boxed{2n+3 \geq \log n}$

$\Rightarrow F(n) = \Omega(n)$

∵ Nearest one is useful


Growth of a fu^n
Time Complexity
f(n)
C.g(n).
Input data.



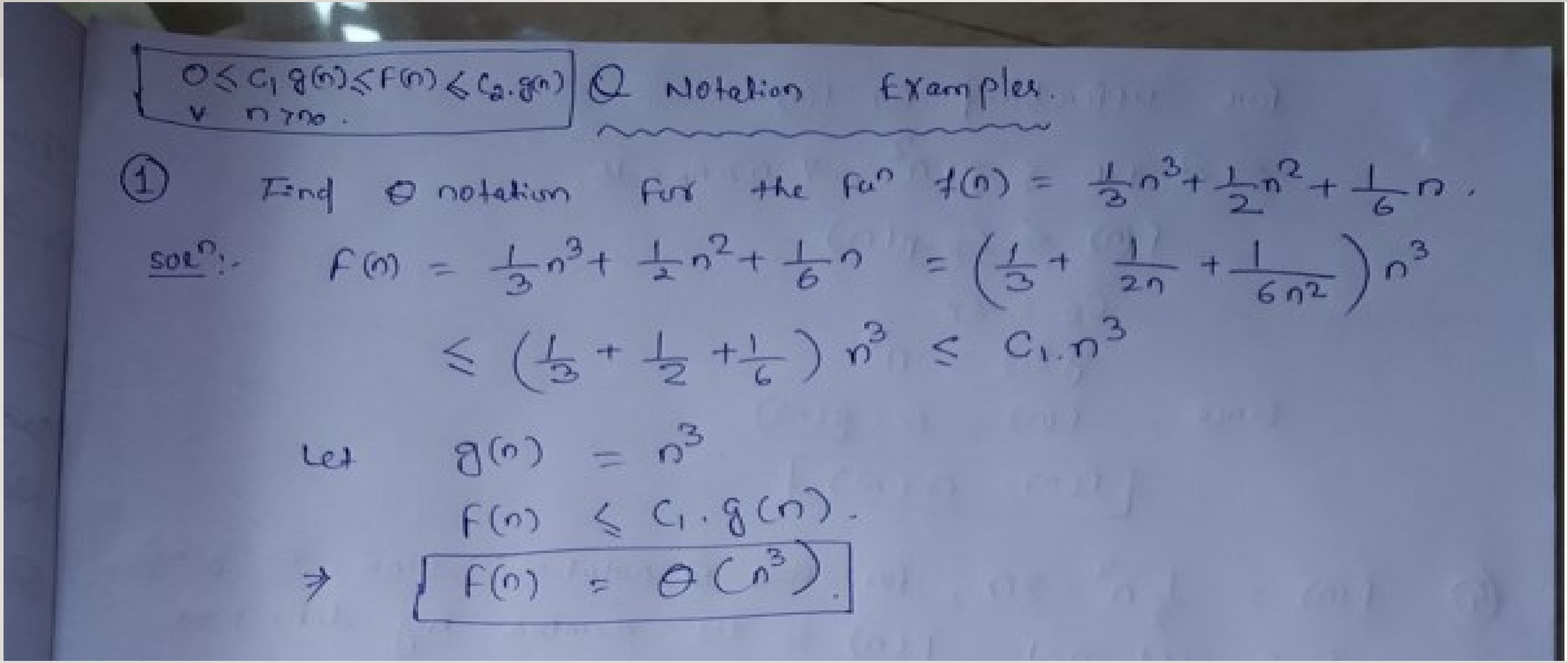
## Θ Notation:-

The function $f(n) = \Theta(g(n))$: there exist positive Constant $c_1, c_2$ and $c_0$ such that

$$0 \le c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n) \text{ for all } n \ge n_0.$$

**Q. Find Θ notation for the function**

**$F(n) = (1/3)n^3 + (1/2)n^2 + (1/6)\ n$**



$0 \leq c_1 g(n) \leq f(n) \leq c_2 . g(n)$
$\forall\ n \geq n_0$.

Θ Notation Examples.

① Find Θ notation for the fcn $f(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$.

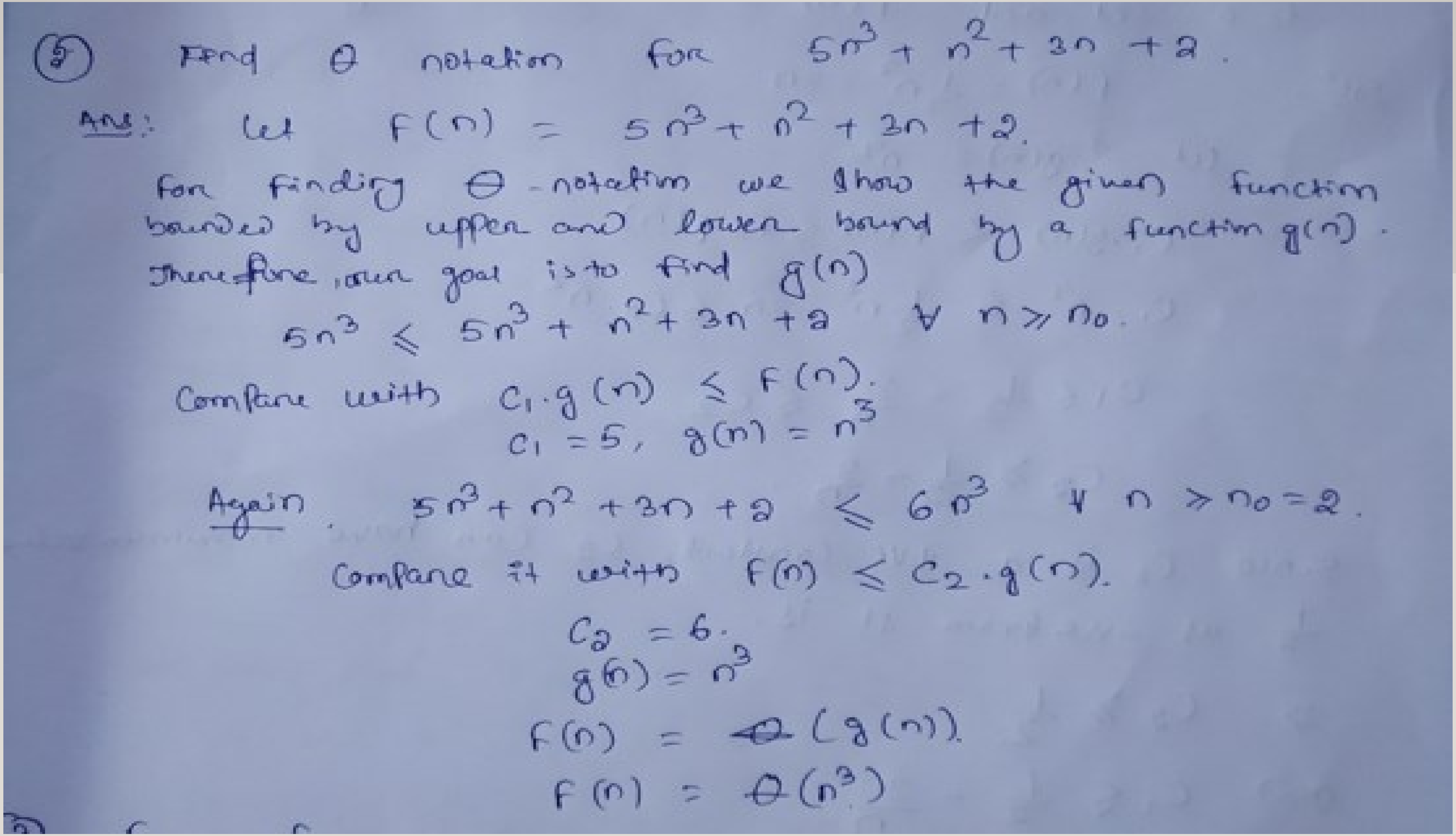
Soln:- $f(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n = \left(\frac{1}{3} + \frac{1}{2n} + \frac{1}{6n^2}\right)n^3$

$\leq \left(\frac{1}{3} + \frac{1}{2} + \frac{1}{6}\right) n^3 \leq c_1 . n^3$

Let $g(n) = n^3$

$f(n) \leq c_1 . g(n)$.

$\Rightarrow \boxed{f(n) = \Theta(n^3)}$.



⑤ Find θ notation for $5n^3 + n^2 + 3n + 2$.

**Ans:** let $f(n) = 5n^3 + n^2 + 3n + 2$.

For finding θ-notation we show the given function bounded by upper and lower bound by a function g(n). Therefore, our goal is to find g(n)

$$5n^3 \leqslant 5n^3 + n^2 + 3n + 2 \quad \forall\ n \geqslant n_0.$$

Compare with $c_1 \cdot g(n) \leqslant f(n)$.

$$c_1 = 5, \quad g(n) = n^3$$

Again. $5n^3 + n^2 + 3n + 2 \leqslant 6n^3 \quad \forall\ n \geqslant n_0 = 2$.

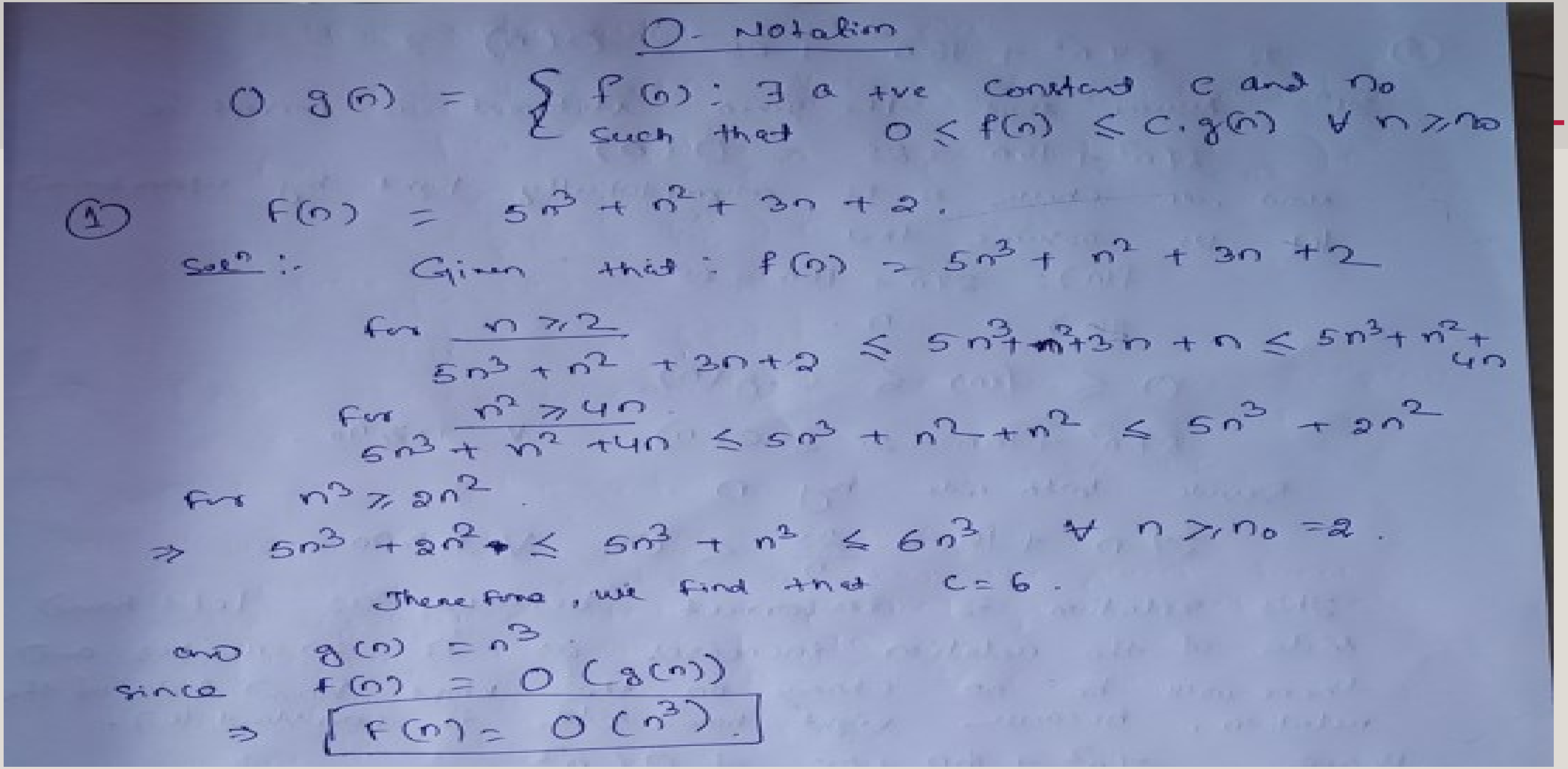
Compare it with $f(n) \leqslant c_2 \cdot g(n)$.

$$c_2 = 6.$$

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

$$f(n) = \theta(g(n)).$$

$$f(n) = \theta(n^3)$$



## O- Notation

$O\,g(n) = \{ f(n) : \exists$ a +ve constant $c$ and $n_0$ such that $0 \leq f(n) \leq c.g(n)\ \forall\ n \geq n_0$

① $f(n) = 5n^3 + n^2 + 3n + 2.$

Solⁿ : Given that : $f(n) = 5n^3 + n^2 + 3n + 2$

for $\underline{n \geq 2}$

$5n^3 + n^2 + 3n + 2 \leq 5n^3 + n^2 + 3n + n \leq 5n^3 + n^2 + 4n$

for $\underline{n^2 \geq 4n}$

$5n^3 + n^2 + 4n \leq 5n^3 + n^2 + n^2 \leq 5n^3 + 2n^2$

for $n^3 \geq 2n^2$.

$\Rightarrow\ 5n^3 + 2n^2 \leq 5n^3 + n^3 \leq 6n^3 \quad \forall\ n \geq n_0 = 2.$

Therefore, we find that $c = 6$.

and $g(n) = n^3$.

since $f(n) = O(g(n))$

$\Rightarrow \boxed{f(n) = O(n^3)}$

Q. Find O notation for the function

$f(n) = 2^n + 6n^2 + 3n$

Solution:

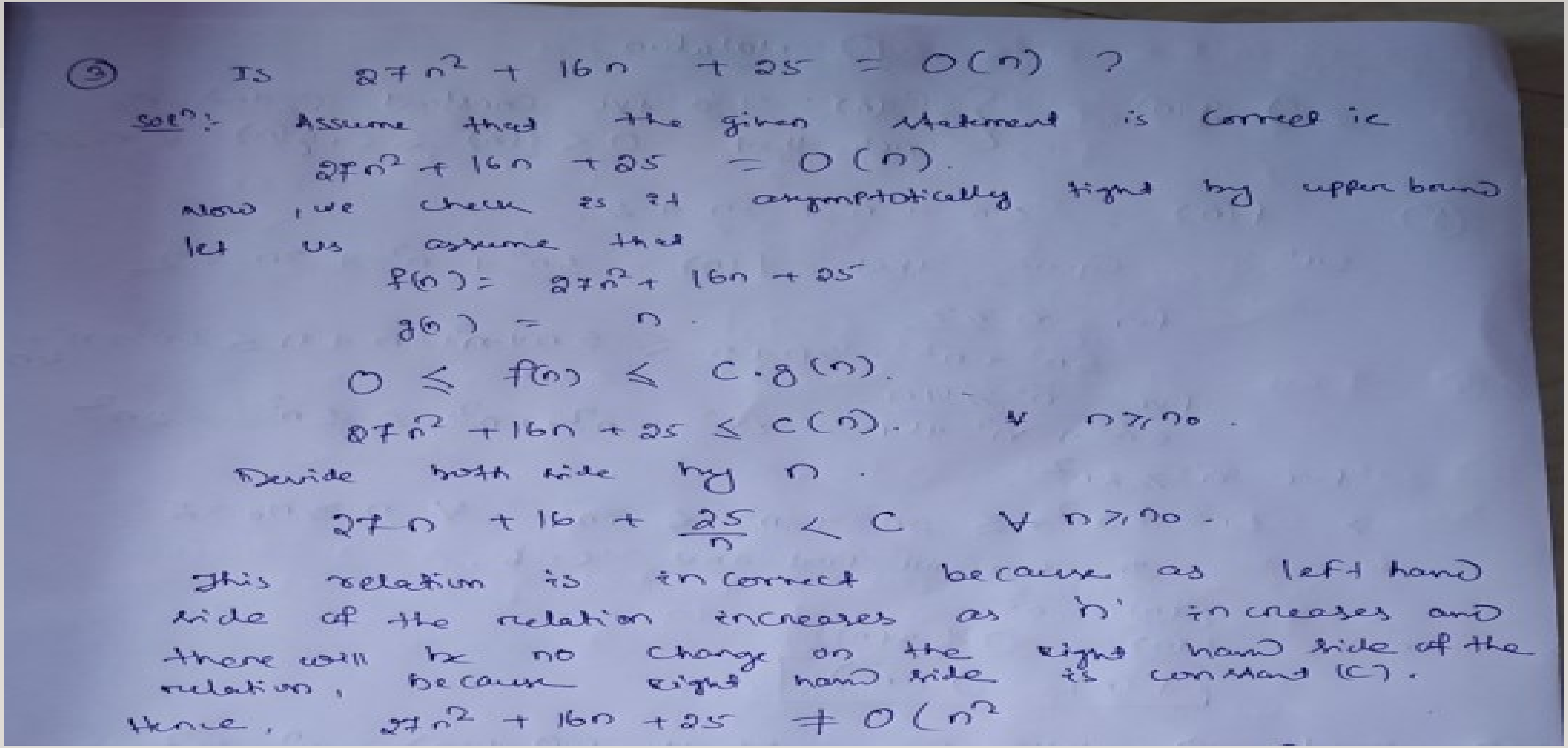
$$2^n + 6n^2 + 3n <= 2^n + 6n^2 + n^2 <= 2^n + 7n^2$$

i.e we are showing the function is asymptotically tight by upper bound for $2^n >= n^2$

Where $n >= 4$

$2^n + 7n^2 <= 2^n + 7n^2 * 2^n$

## Q. Is $27n^2 + 16n + 25 = O(n)$ ?



③ Is $27n^2 + 16n + 25 = O(n)$ ?

Solⁿ:- Assume that the given statement is correct ie

$$27n^2 + 16n + 25 = O(n).$$

Now, we check is it asymptotically tight by upper bound

let us assume that

$$f(n) = 27n^2 + 16n + 25$$

$$g(n) = n.$$

$$0 \leqslant f(n) \leqslant c \cdot g(n).$$

$$27n^2 + 16n + 25 \leq c(n). \quad \forall \; n \geqslant n_0.$$

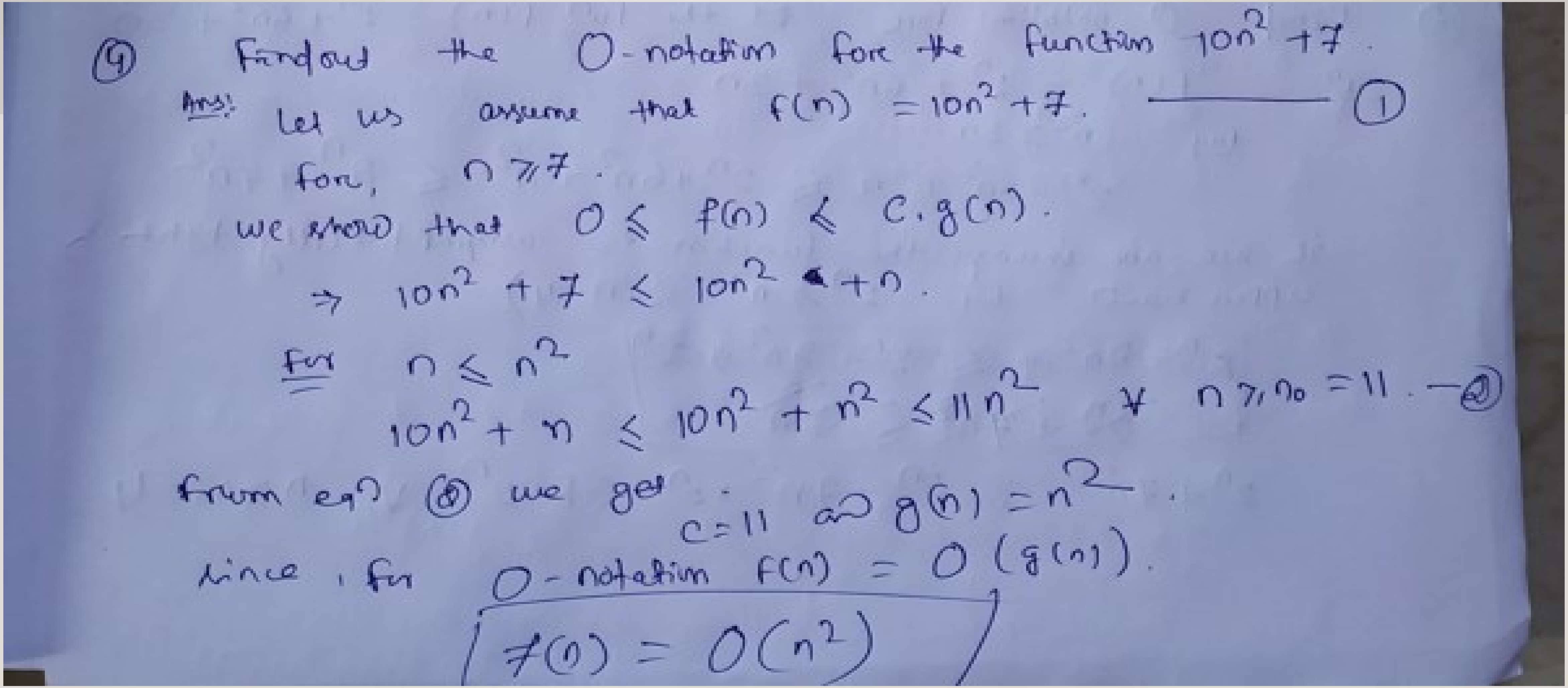
Devide both side by $n$.

$$27n + 16 + \frac{25}{n} < c \quad \forall \; n \geqslant n_0.$$

This relation is incorrect because as left hand side of the relation increases as 'n' increases and there will be no change on the right hand side of the relation, because right hand side is constant (c).

Hence, $27n^2 + 16n + 25 \neq O(n^2$

# Q. Find out the O notation for the function $10n^2 + 7$



④ Find out the O-notation for the function $10n^2 + 7$.

Ans: Let us assume that $f(n) = 10n^2 + 7$. ——— ①

for, $n \geq 7$.

we show that $0 \leq f(n) \leq c.g(n)$.

$\Rightarrow 10n^2 + 7 \leq 10n^2 + n$.

for $n \leq n^2$

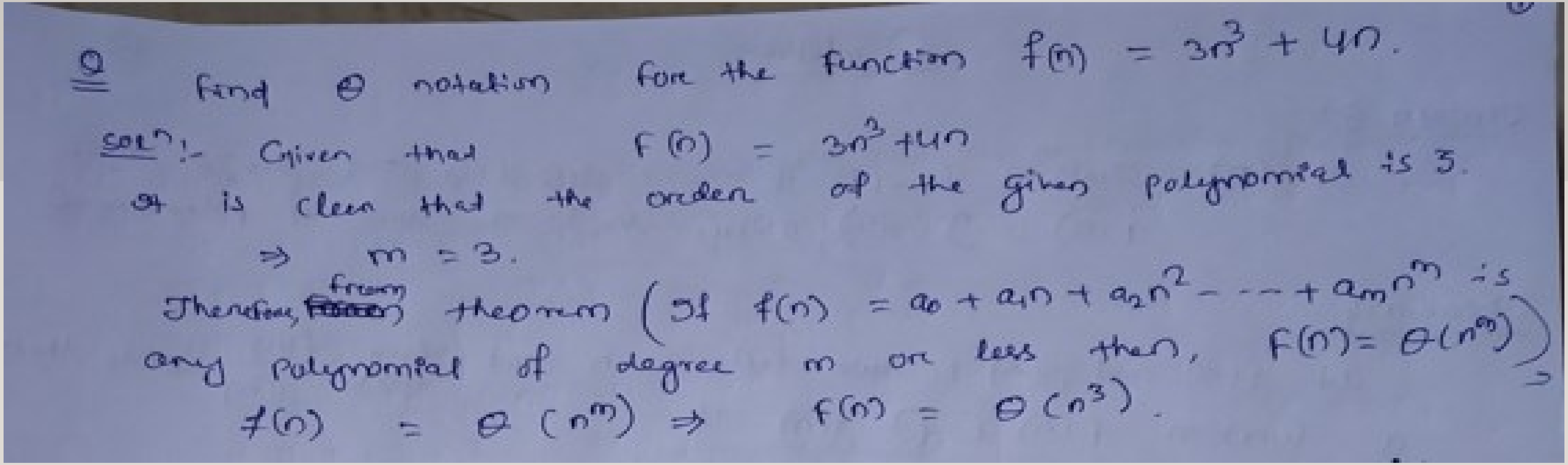
$10n^2 + n \leq 10n^2 + n^2 \leq 11n^2$ ∀ $n \geq n_0 = 11$. — ②

from eqn ② we get.

$c = 11$ an $g(n) = n^2$.

since, for O-notation $f(n) = O(g(n))$.

$$\boxed{f(n) = O(n^2)}$$



Q. Find Θ notation for the function $f(n) = 3n^3 + 4n$.

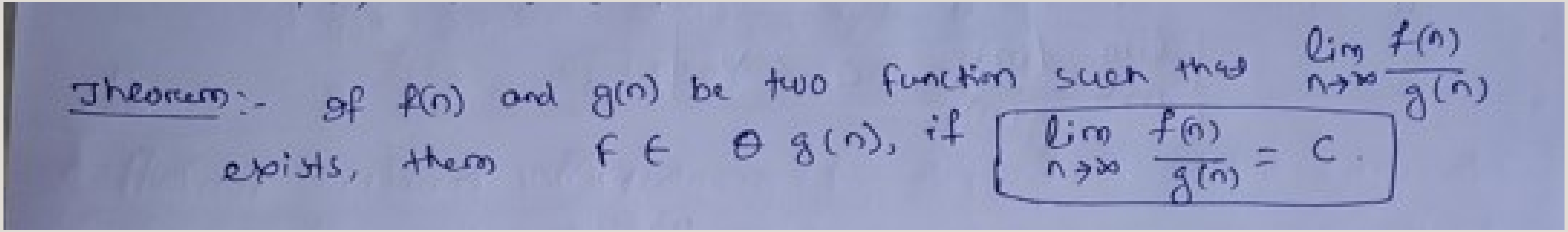
Soln:- Given that $f(n) = 3n^3 + 4n$

It is clear that the order of the given polynomial is 3.

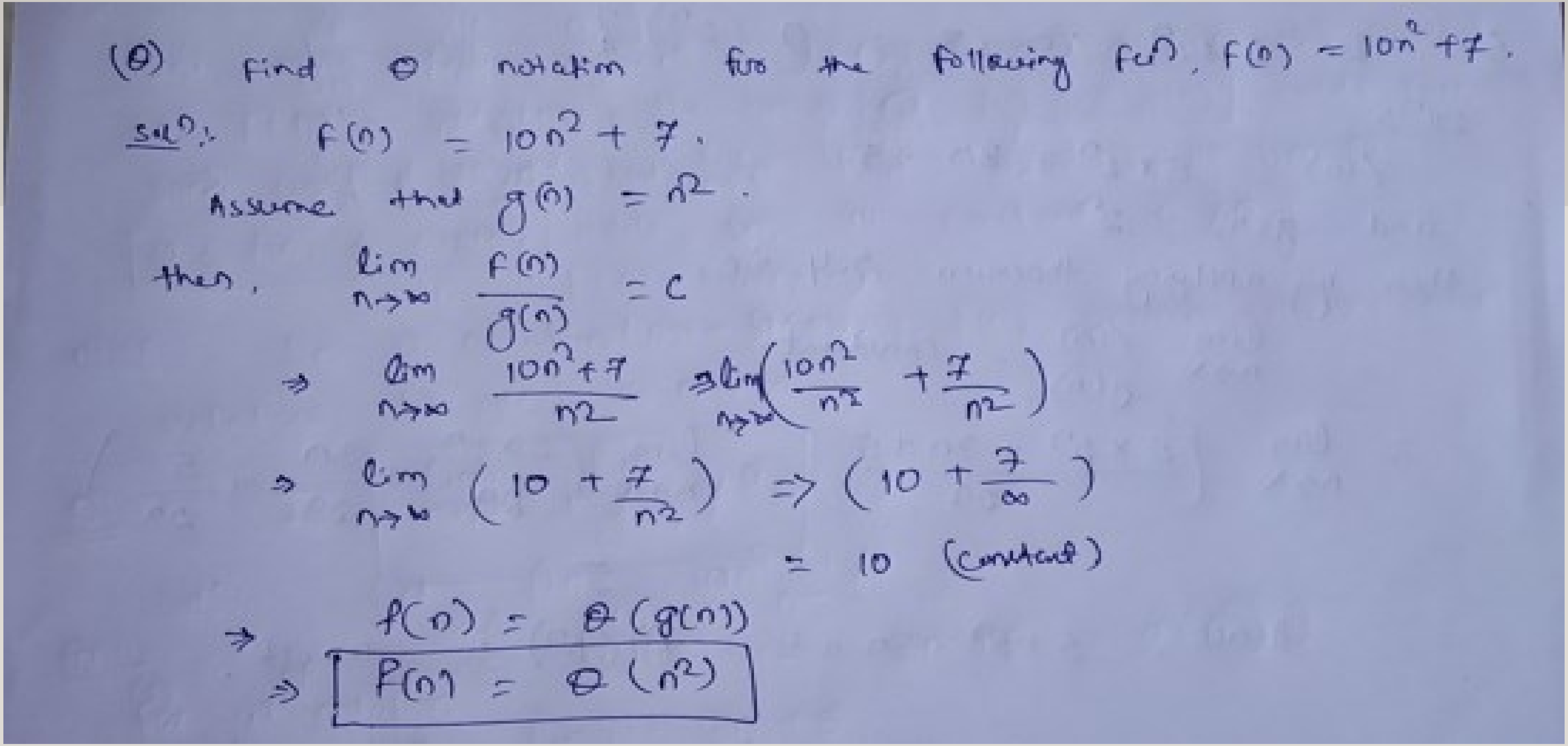
$\Rightarrow m = 3$.

Therefore, from theorem (If $f(n) = a_0 + a_1 n + a_2 n^2 - - - + a_m n^m$ is any polynomial of degree m or less then, $F(n) = \Theta(n^m)$)

$\Rightarrow f(n) = \Theta(n^m) \Rightarrow f(n) = \Theta(n^3)$.



Theorem:- If f(n) and g(n) be two function such that $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists, then $f \in \Theta\, g(n)$, if $\boxed{\lim_{n\to\infty} \frac{f(n)}{g(n)} = C}$.



(Θ) Find Θ notation for the following fc$^n$. $f(n) = 10n^2 + 7$.

Sol$^n$: $f(n) = 10n^2 + 7$.

Assume that $g(n) = n^2$.

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

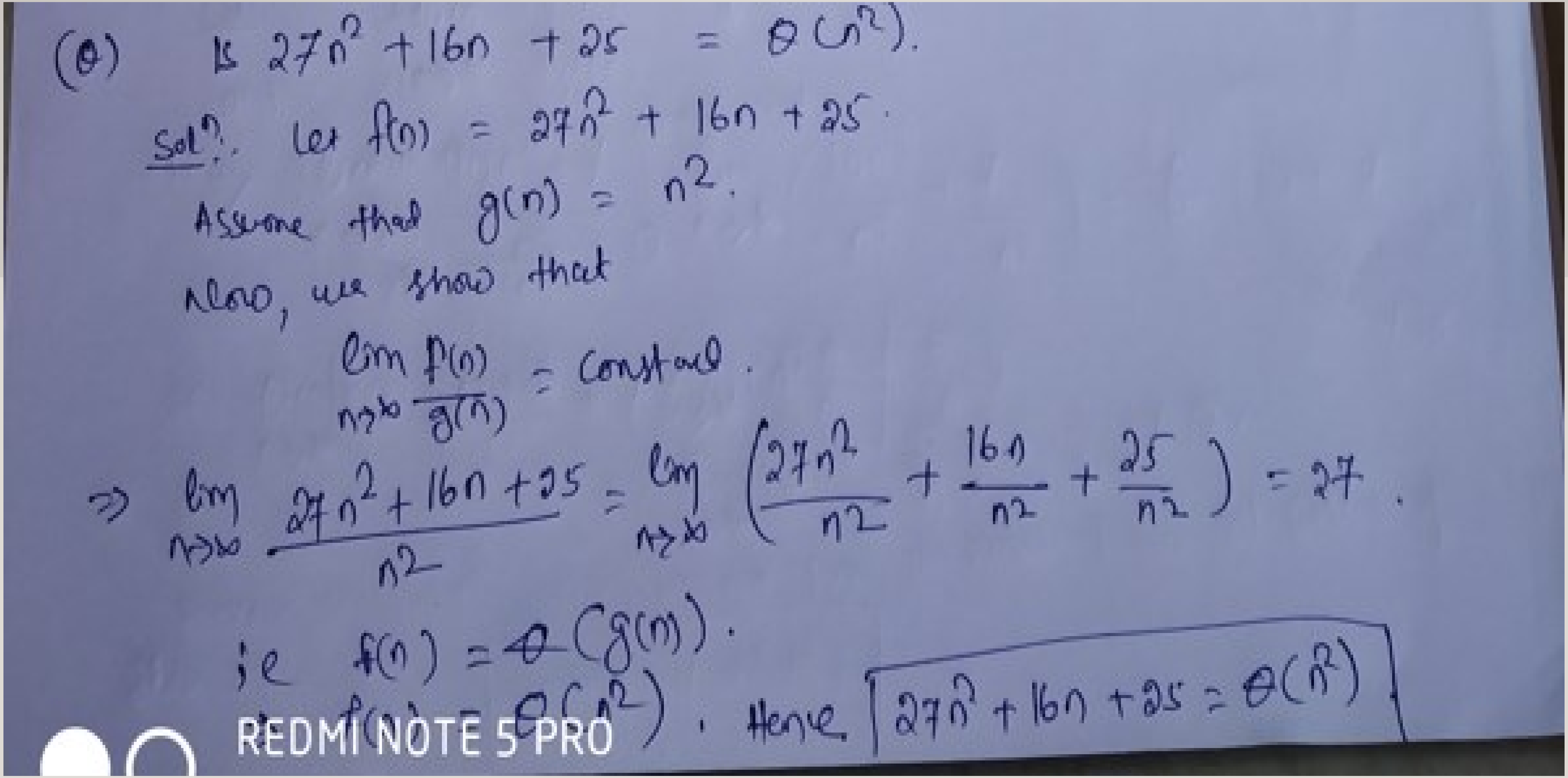
$\Rightarrow \lim_{n\to\infty} \frac{10n^2+7}{n^2} \Rightarrow \lim_{n\to\infty}\left(\frac{10n^2}{n^2} + \frac{7}{n^2}\right)$

$\Rightarrow \lim_{n\to\infty}\left(10 + \frac{7}{n^2}\right) \Rightarrow \left(10 + \frac{7}{\infty}\right)$

$= 10$ (constant)

$\Rightarrow f(n) = \Theta(g(n))$

$\Rightarrow \boxed{f(n) = \Theta(n^2)}$



(Q) Is $27n^2 + 16n + 25 = \Theta(n^2)$.

Sol$^n$: Let $f(n) = 27n^2 + 16n + 25$.

Assume that $g(n) = n^2$.

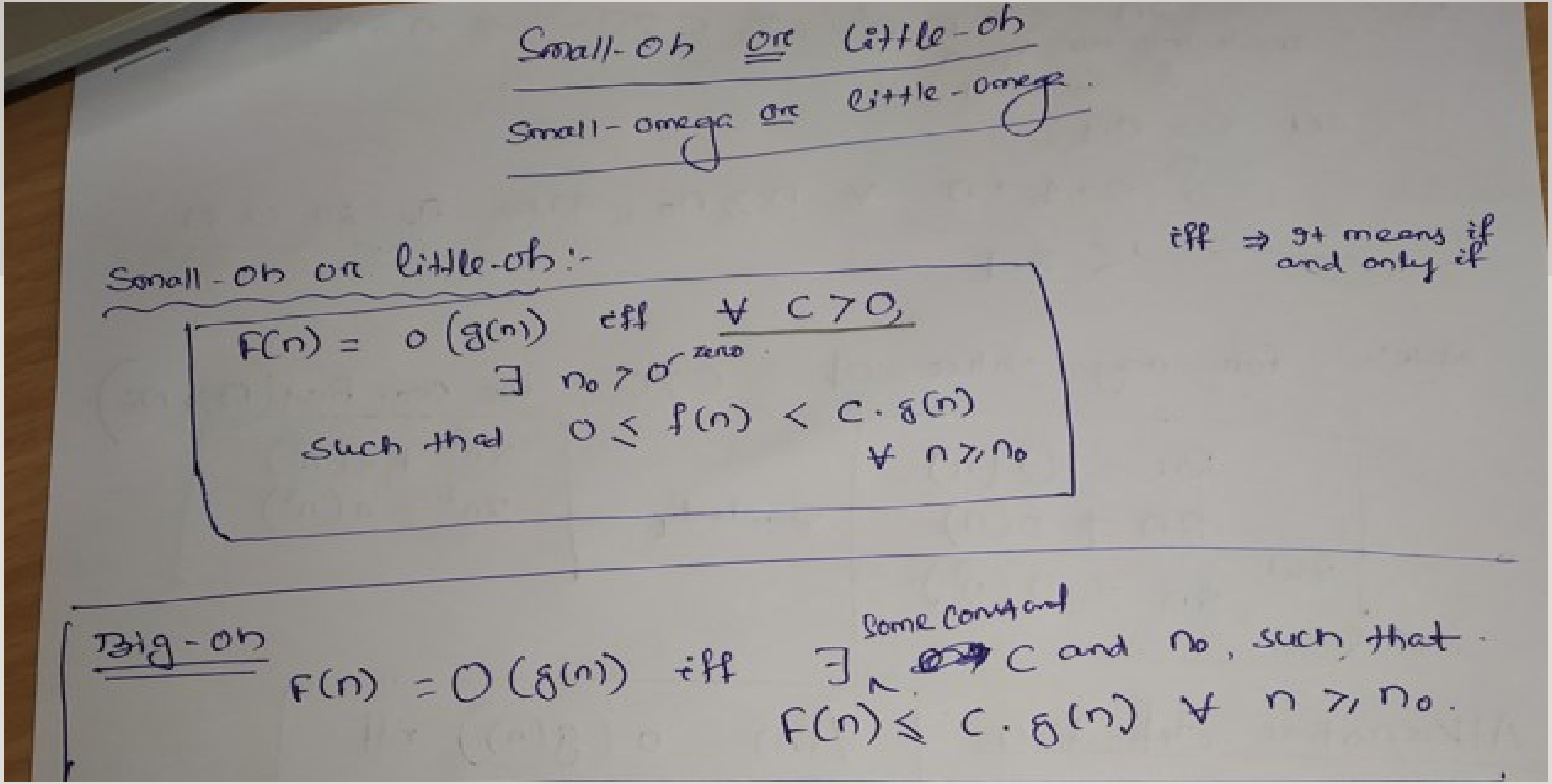
Now, we show that

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \text{constant}.$$

$$\Rightarrow \lim_{n\to\infty} \frac{27n^2 + 16n + 25}{n^2} = \lim_{n\to\infty} \left( \frac{27n^2}{n^2} + \frac{16n}{n^2} + \frac{25}{n^2} \right) = 27.$$

i.e $f(n) = \Theta(g(n))$.

$f(n) = \Theta(n^2)$. Hence $\boxed{27n^2 + 16n + 25 = \Theta(n^2)}$



## Small-oh or little-oh

## Small-omega or little-omega.

iff ⇒ It means if and only if

**Small-oh or little-oh :-**

$F(n) = o(g(n))$ iff $\forall\ c > 0,$

$\exists\ n_0 > \overset{\text{zero}}{0}$

such that $0 \leq f(n) < c \cdot g(n)$

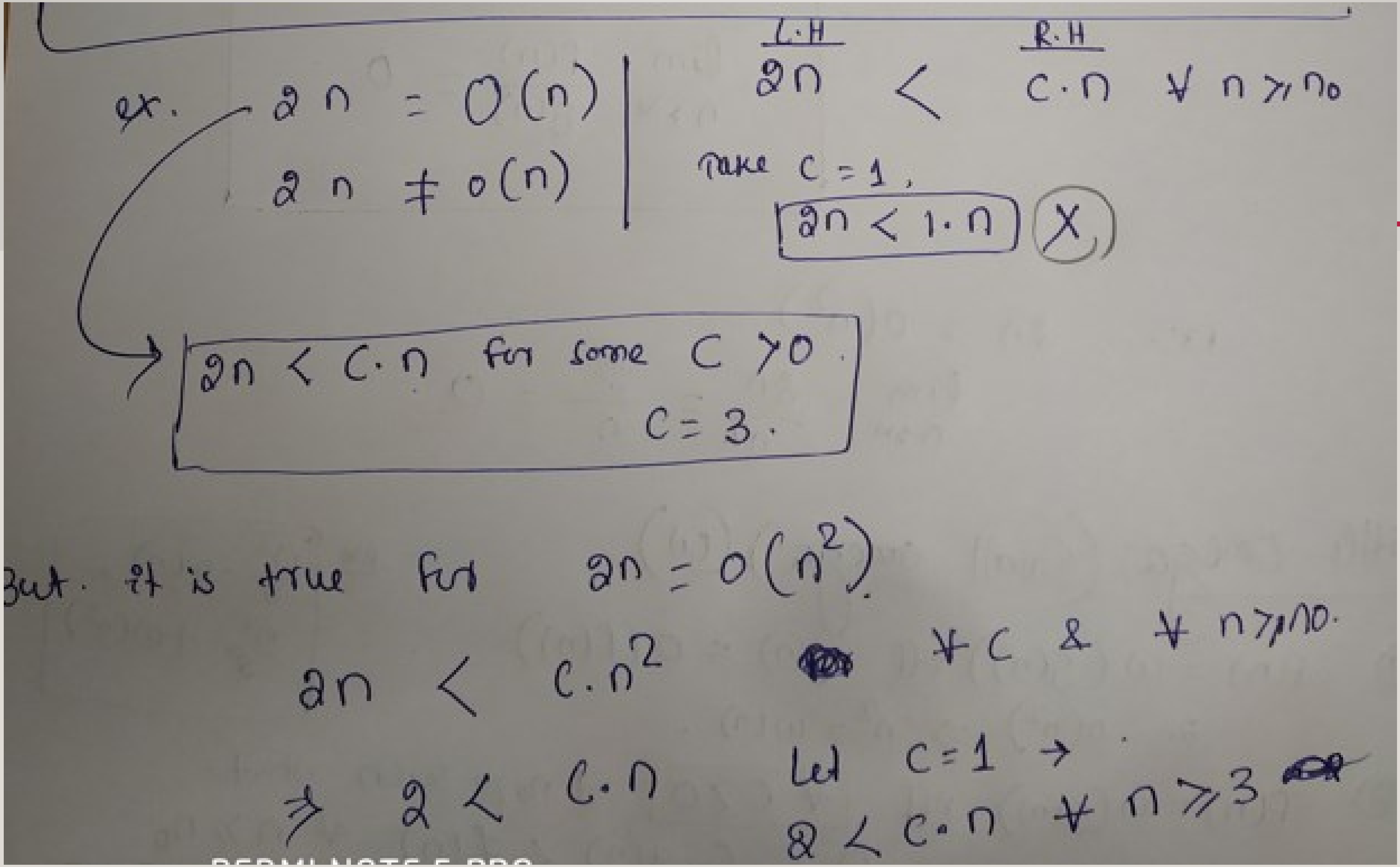
$\forall\ n \geq n_0$

---

**Big-oh**

$F(n) = O(g(n))$ iff $\exists$ some constant $c$ and $n_0$, such that

$F(n) \leq c \cdot g(n)\ \forall\ n \geq n_0$.



ex. $2n = O(n)$

$2n \neq o(n)$

$\overset{L.H}{2n} < \overset{R.H}{c\cdot n} \;\forall\; n \geq n_0$

take $c = 1$,

$\boxed{2n < 1\cdot n}$ X

$\boxed{2n < c\cdot n \text{ for some } c > 0. \quad c = 3.}$

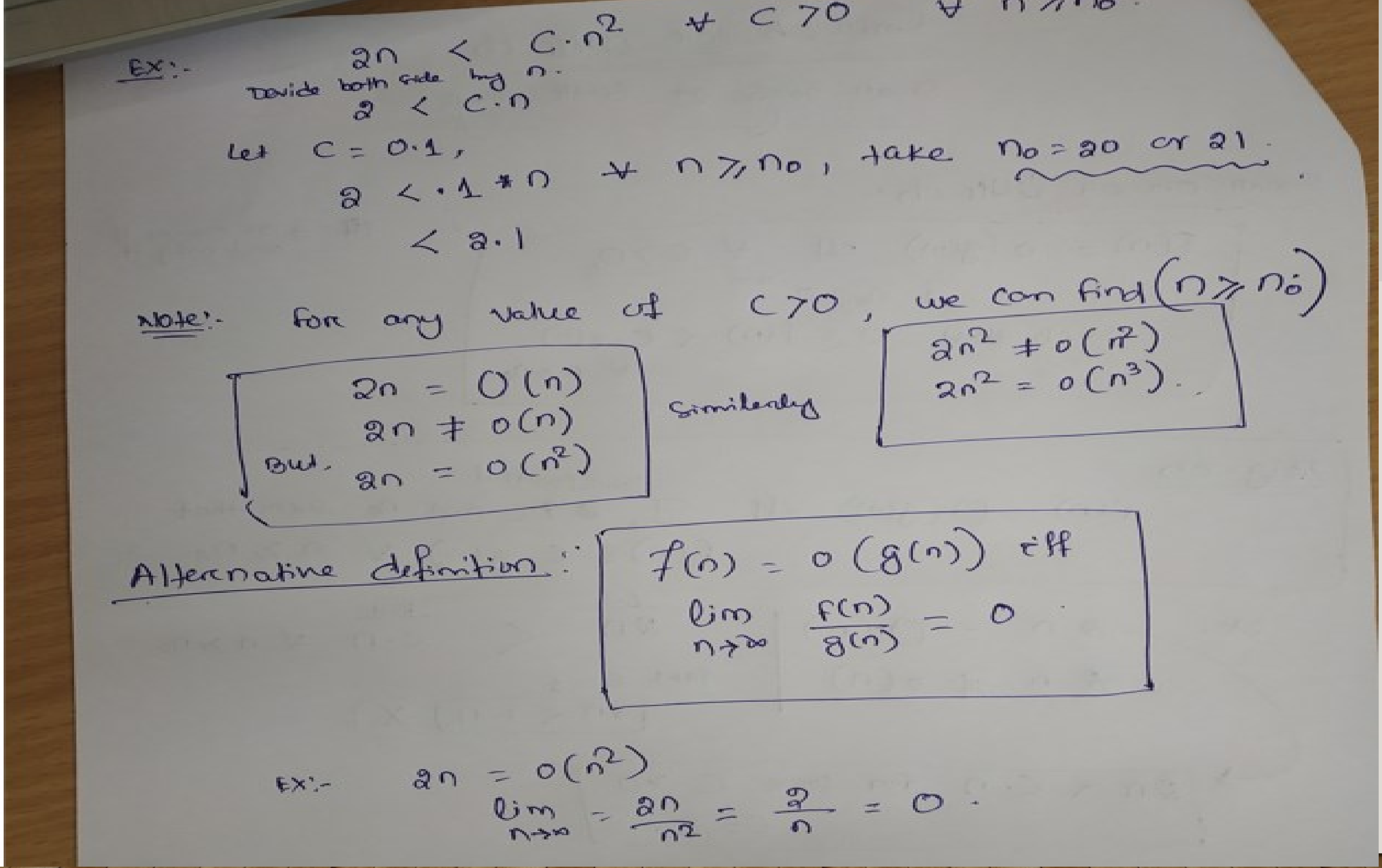
But. it is true for $2n = o(n^2)$.

$2n < c\cdot n^2 \;\forall c \;\&\; \forall\; n \geq n_0.$

$\Rightarrow 2 < c\cdot n$

Let $c = 1 \rightarrow$

$2 < c\cdot n \;\forall\; n \geq 3$



Ex:- $2n < C\cdot n^2 \quad \forall\, C > 0 \quad \forall\, n \geq n_0$.

Devide both side by n.

$2 < C\cdot n$

Let $C = 0.1$,

$2 < .1 * n \quad \forall\, n \geq n_0$, take $n_0 = 20$ or $21$.

$< 2.1$

**Note**:- for any value of $C > 0$, we can find $(n \geq n_0)$

$2n = O(n)$

$2n \neq o(n)$

But, $2n = o(n^2)$

Similarly

$2n^2 \neq o(n^2)$

$2n^2 = o(n^3)$.

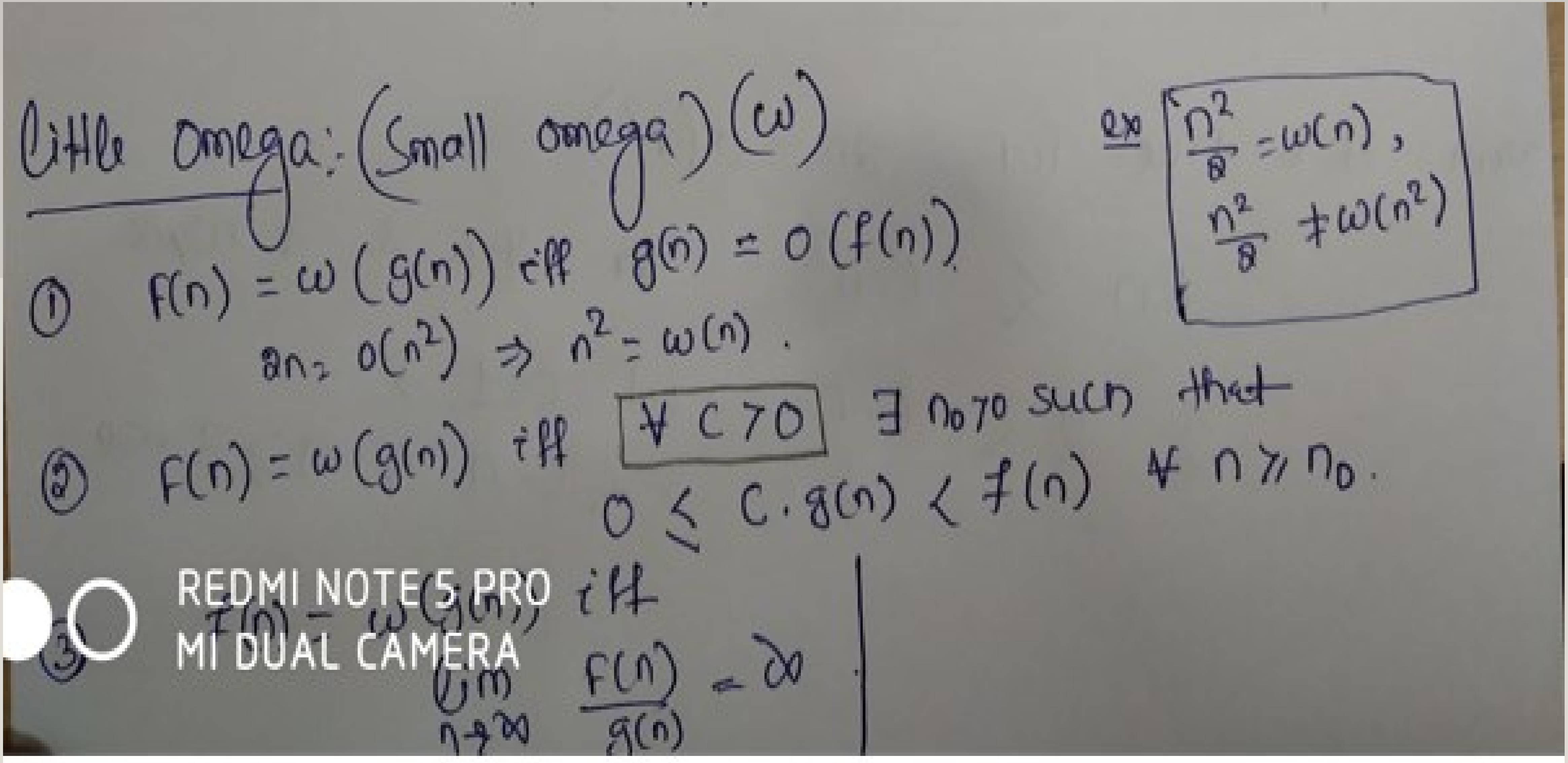
Alternative definition:

$f(n) = o(g(n))$ iff

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Ex:- $2n = o(n^2)$

$$\lim_{n\to\infty} = \frac{2n}{n^2} = \frac{2}{n} = 0.$$



Little omega: (Small omega) ($\omega$)

ex $\boxed{\frac{n^2}{8} = \omega(n), \quad \frac{n^2}{8} \neq \omega(n^2)}$

① $f(n) = \omega(g(n))$ iff $g(n) = o(f(n))$

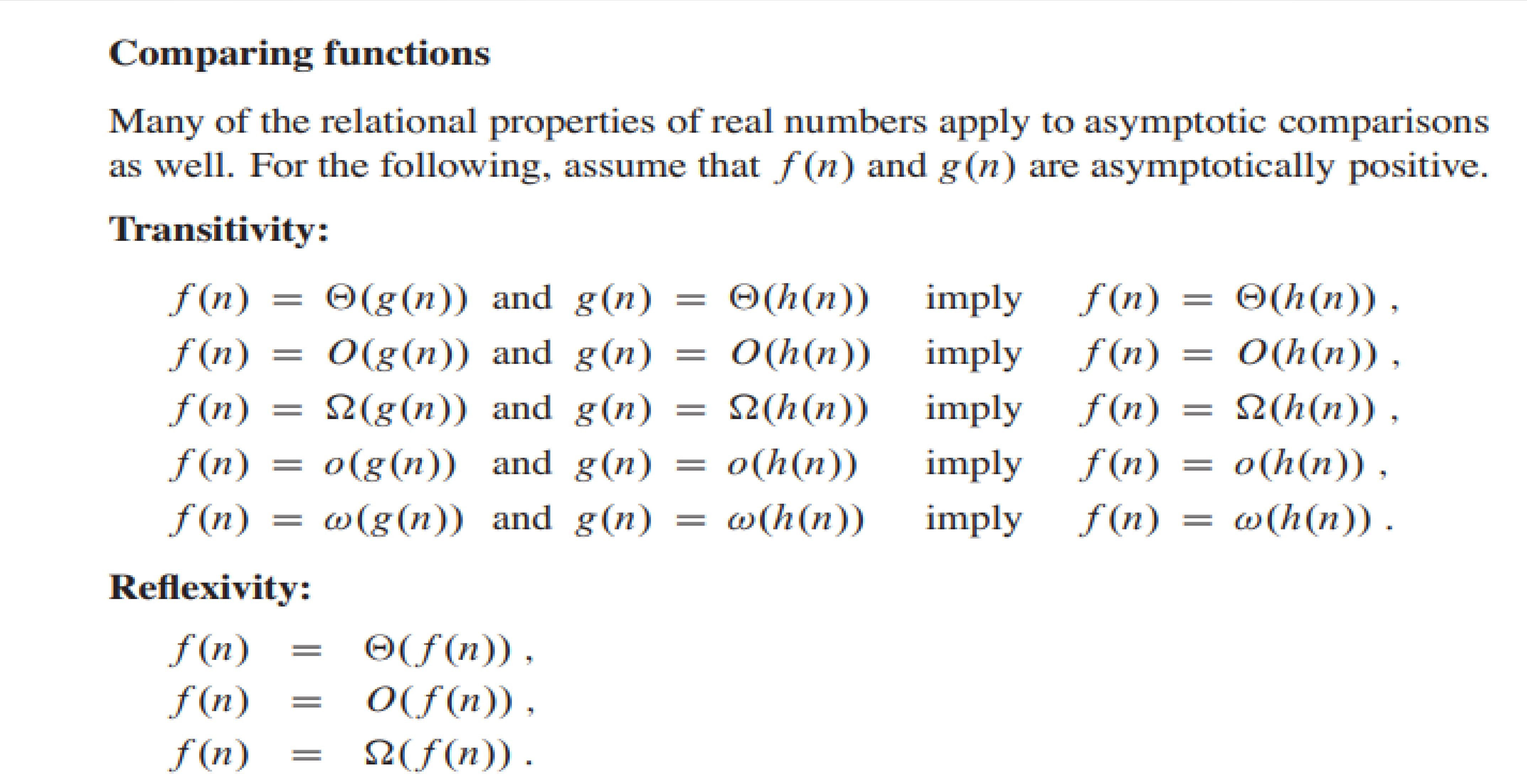
$8n = o(n^2) \Rightarrow n^2 = \omega(n)$.

② $f(n) = \omega(g(n))$ iff $\boxed{\forall c > 0}$ $\exists n_0 > 0$ such that

$0 \leq c \cdot g(n) < f(n) \quad \forall n \geq n_0$.

③ $f(n) = \omega(g(n))$ iff

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



## Comparing functions

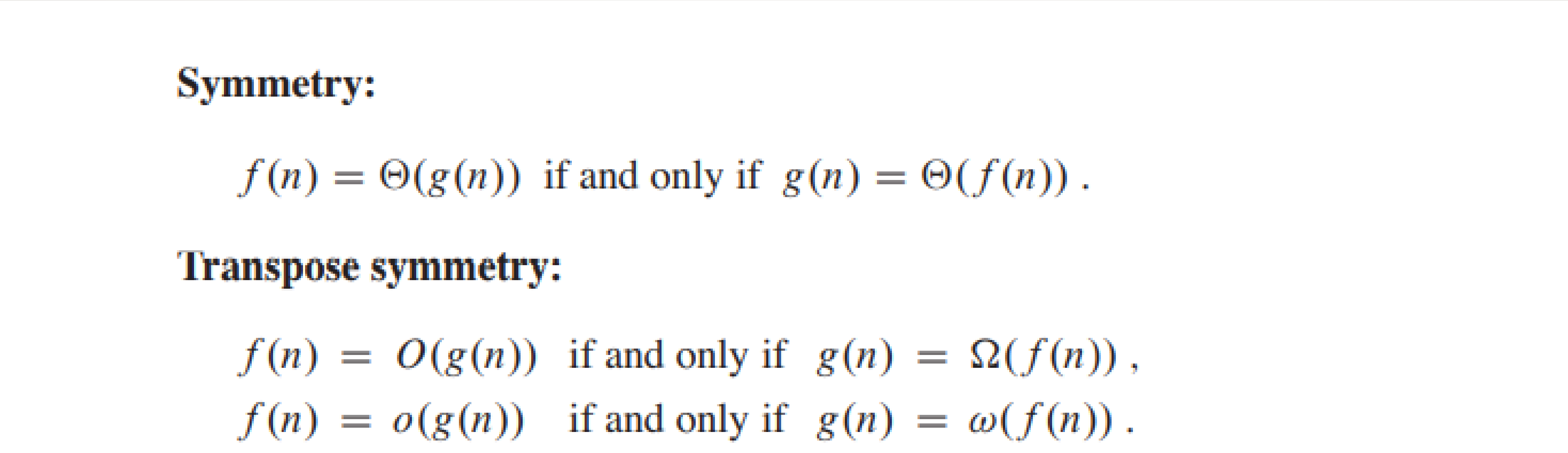
Many of the relational properties of real numbers apply to asymptotic comparisons as well. For the following, assume that $f(n)$ and $g(n)$ are asymptotically positive.

**Transitivity:**

$$\begin{array}{lllll}
f(n) = \Theta(g(n)) & \text{and} & g(n) = \Theta(h(n)) & \text{imply} & f(n) = \Theta(h(n)) \,, \\
f(n) = O(g(n)) & \text{and} & g(n) = O(h(n)) & \text{imply} & f(n) = O(h(n)) \,, \\
f(n) = \Omega(g(n)) & \text{and} & g(n) = \Omega(h(n)) & \text{imply} & f(n) = \Omega(h(n)) \,, \\
f(n) = o(g(n)) & \text{and} & g(n) = o(h(n)) & \text{imply} & f(n) = o(h(n)) \,, \\
f(n) = \omega(g(n)) & \text{and} & g(n) = \omega(h(n)) & \text{imply} & f(n) = \omega(h(n)) \,.
\end{array}$$

**Reflexivity:**

$$\begin{aligned}
f(n) &= \Theta(f(n)) \,, \\
f(n) &= O(f(n)) \,, \\
f(n) &= \Omega(f(n)) \,.
\end{aligned}$$



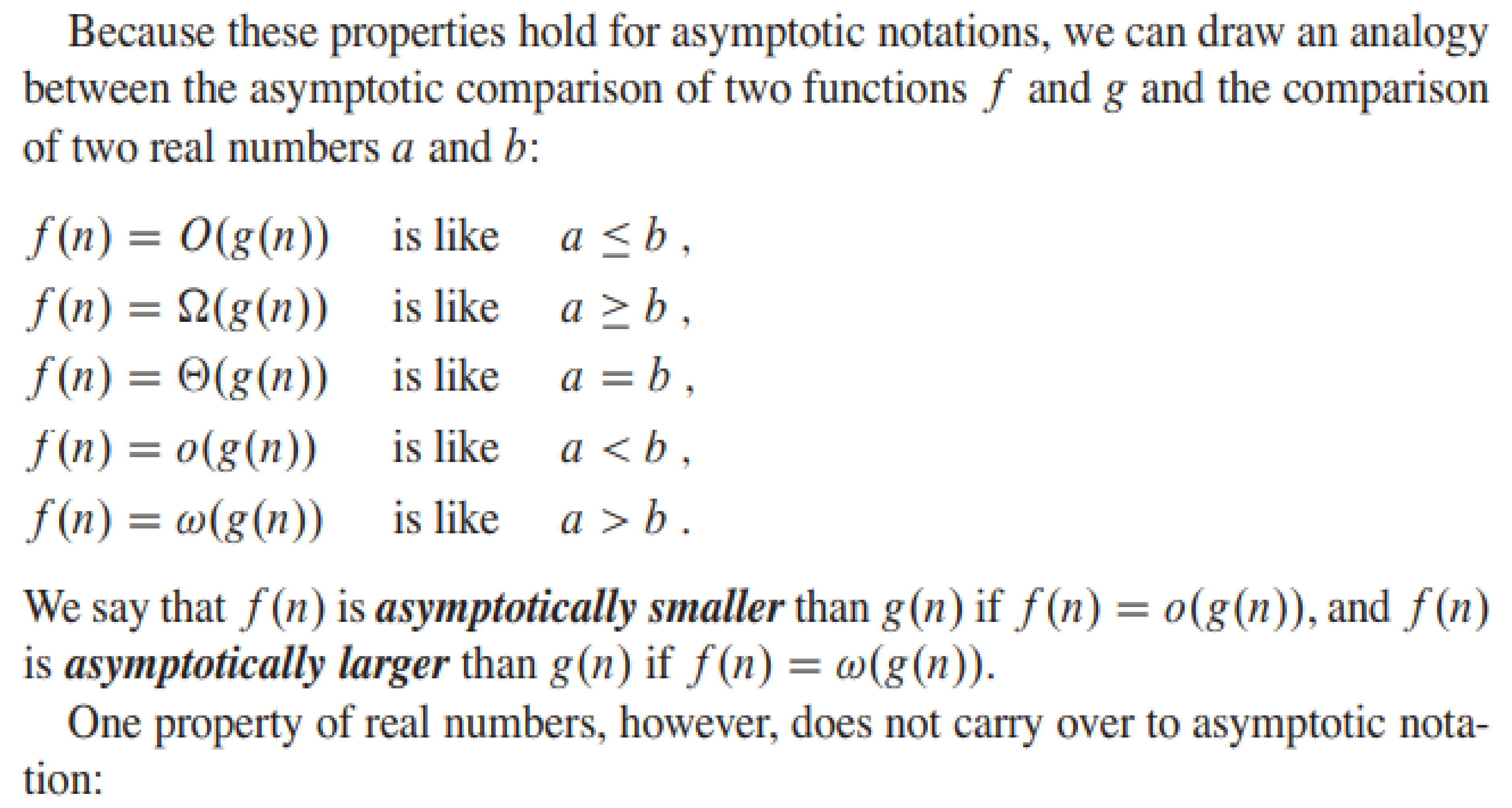
**Symmetry:**

$$f(n) = \Theta(g(n)) \text{ if and only if } g(n) = \Theta(f(n)) .$$

**Transpose symmetry:**

$$f(n) = O(g(n)) \text{ if and only if } g(n) = \Omega(f(n)) ,$$

$$f(n) = o(g(n)) \text{ if and only if } g(n) = \omega(f(n)) .$$



Because these properties hold for asymptotic notations, we can draw an analogy between the asymptotic comparison of two functions $f$ and $g$ and the comparison of two real numbers $a$ and $b$:

$$\begin{aligned}
f(n) &= O(g(n)) && \text{is like} && a \le b \,, \\
f(n) &= \Omega(g(n)) && \text{is like} && a \ge b \,, \\
f(n) &= \Theta(g(n)) && \text{is like} && a = b \,, \\
f(n) &= o(g(n)) && \text{is like} && a < b \,, \\
f(n) &= \omega(g(n)) && \text{is like} && a > b \,.
\end{aligned}$$

We say that $f(n)$ is ***asymptotically smaller*** than $g(n)$ if $f(n) = o(g(n))$, and $f(n)$ is ***asymptotically larger*** than $g(n)$ if $f(n) = \omega(g(n))$.

One property of real numbers, however, does not carry over to asymptotic notation:

- Reflexivity Example:

If $f(n) = n^3 \Rightarrow O(n^3)$

**Symmetry**:

If $f(n) = n^2$ and $g(n) = n^2$ then $f(n) = \Theta(n^2)$ and $g(n) = \Theta(n^2)$

- ***Necessary part:***

  $f(n) = \Theta(g(n)) \Rightarrow g(n) = \Theta(f(n))$

  By the definition of $\Theta$, there exists positive constants $c1$, $c2$, $no$ such that

  $c1.g(n) \leq f(n) \leq c2.g(n)$ for all $n \geq no$

  $\Rightarrow g(n) \leq (1/c1).f(n)$ and $g(n) \geq (1/c2).f(n)$

  $\Rightarrow (1/c2).f(n) \leq g(n) \leq (1/c1).f(n)$

  Since $c1$ and $c2$ are positive constants, $1/c1$ and $1/c2$ are well defined.

  Therefore, by the definition of $\Theta$, $g(n) = \Theta(f(n))$

- ***Sufficiency part:***

  $g(n) = \Theta(f(n)) \Rightarrow f(n) = \Theta(g(n))$

  By the definition of $\Theta$, there exists positive constants $c1$, $c2$, $no$ such that

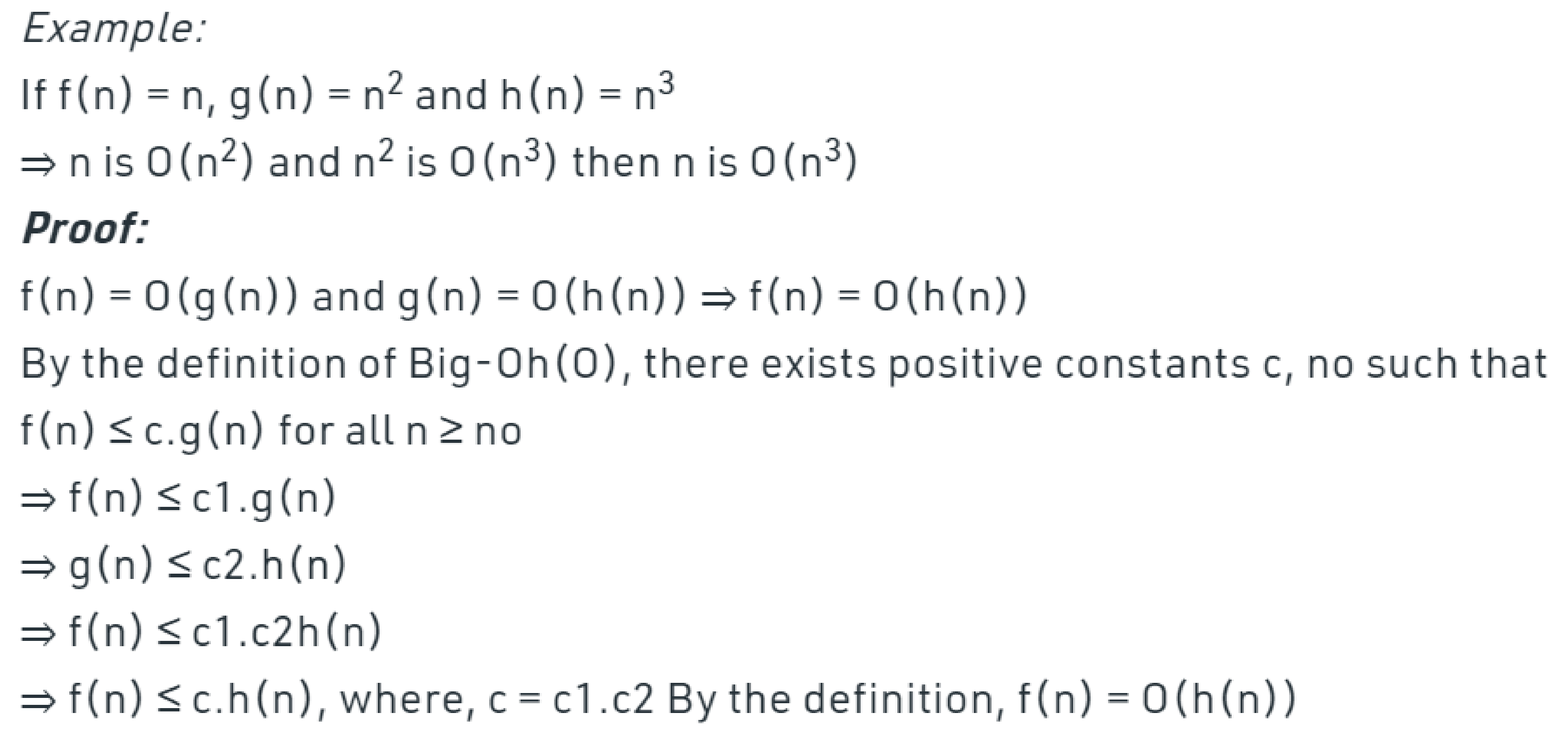
  $c1.f(n) \leq g(n) \leq c2.f(n)$ for all $n \geq no$

  $\Rightarrow f(n) \leq (1/c1).g(n)$ and $f(n) \geq (1/c2).g(n)$

  $\Rightarrow (1/c2).g(n) \leq f(n) \leq (1/c1).g(n)$

  By the definition of Theta($\Theta$), $f(n) = \Theta(g(n))$

- Transitivity Example



*Example:*

If $f(n) = n$, $g(n) = n^2$ and $h(n) = n^3$

$\Rightarrow$ $n$ is $O(n^2)$ and $n^2$ is $O(n^3)$ then $n$ is $O(n^3)$

***Proof:***

$f(n) = O(g(n))$ and $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$

By the definition of Big-Oh(O), there exists positive constants c, no such that

$f(n) \leq c.g(n)$ for all $n \geq no$

$\Rightarrow f(n) \leq c1.g(n)$

$\Rightarrow g(n) \leq c2.h(n)$

$\Rightarrow f(n) \leq c1.c2h(n)$

$\Rightarrow f(n) \leq c.h(n)$, where, $c = c1.c2$ By the definition, $f(n) = O(h(n))$