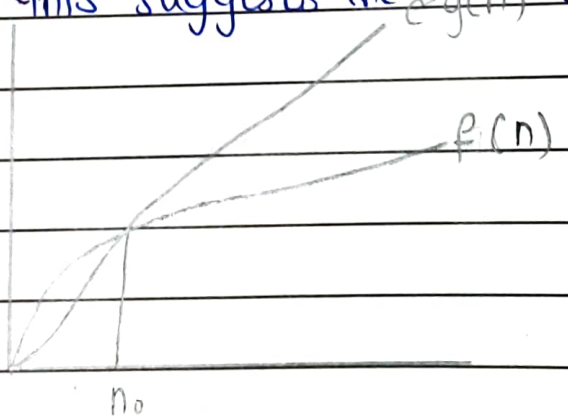
Question Nos.

Q.1] 1] Asymptotic notions are used to check the time etc complexity of algorithms and to find out which algorithm is the most time efficient when input size is large enough.

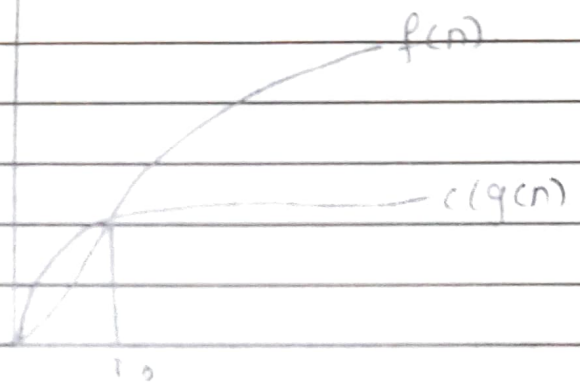
2] Big O → This is used to represent the worst case of the time complexity. This suggests the max time that an algorithm can take for completion



In the above graph $cg(n)$ and $f(n)$ are functions and we can say that $cg(n) = O(f(n))$ if and only if $cg(n) \geq f(n)$ and $n > n_0$.

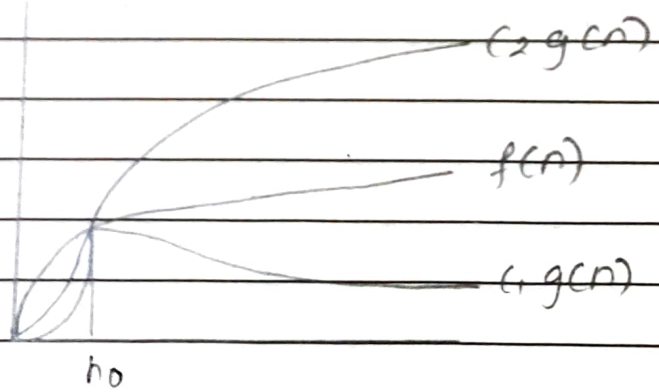
Question Nos. | Marks Awarded

2] Big. Ω → This is used to represent the best of time complexity of an algorithm. This suggests the minimum time algorithm can can ~~fo~~ take for its completion.



From the above graph we can say that ~~f(n) = Ω(c g(n))~~ ~~c g(n) = Ω(f(n))~~ if and only if $f(n) \geq c(g(n))$ and $n > n_0$.

3] Big θ → This is used to represent the average of time complexity of an algorithm. This suggests the average time an algorithm can take for its completion.



From above graph we can say that ~~c(g(n)) = f(n) = θ(c(g(n))~~ ~~c g(n) = θ(f(n))~~

$c(g(n)) = \theta(f(n))$ if and only if

Question Nos.		Marks Awarded

$c_1 g(n) <= f(n) <= c_2 g(n)$ and
$n > n_0$;