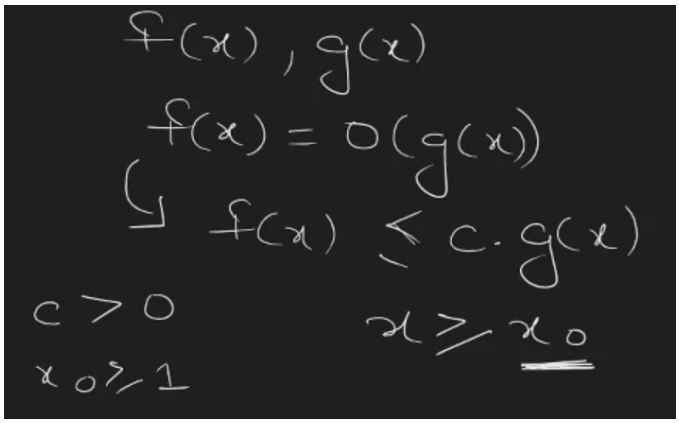
04_Asymptotic Notations

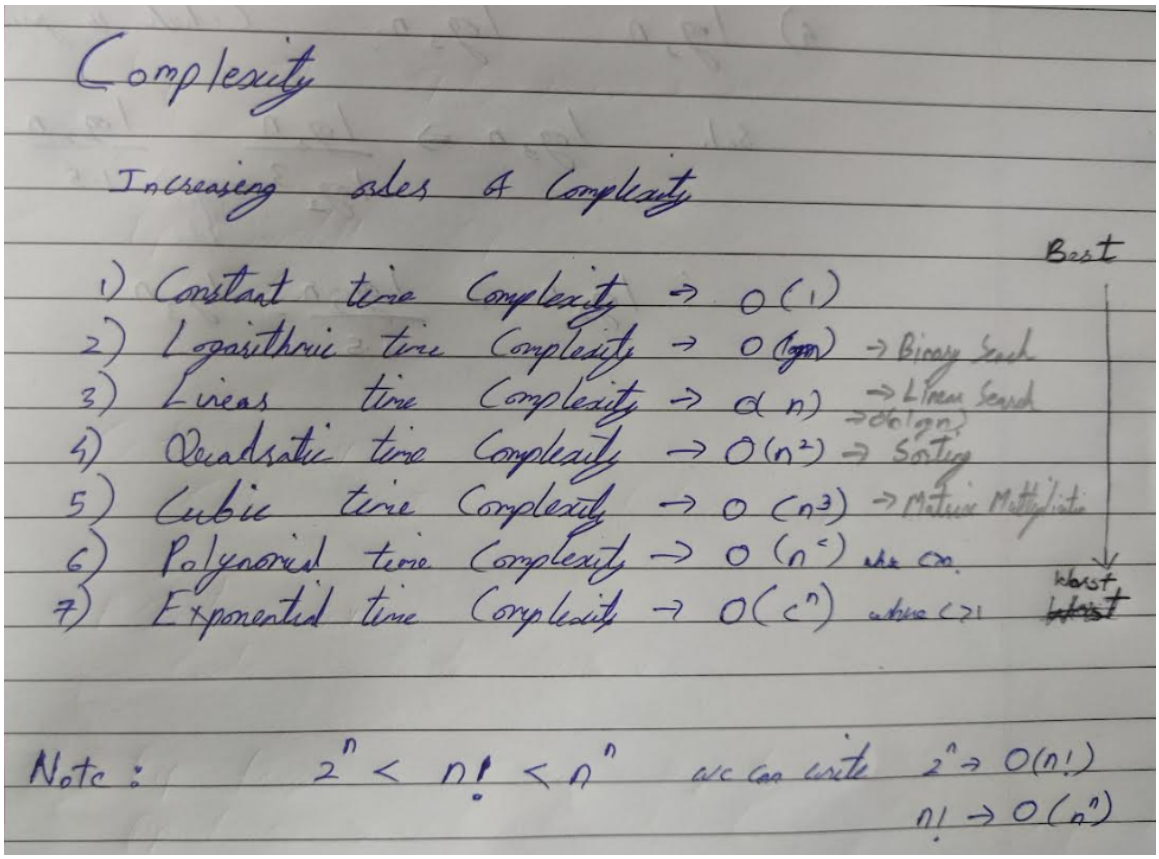
Running time of an Algorithm

Time and space complexity of an algorithm

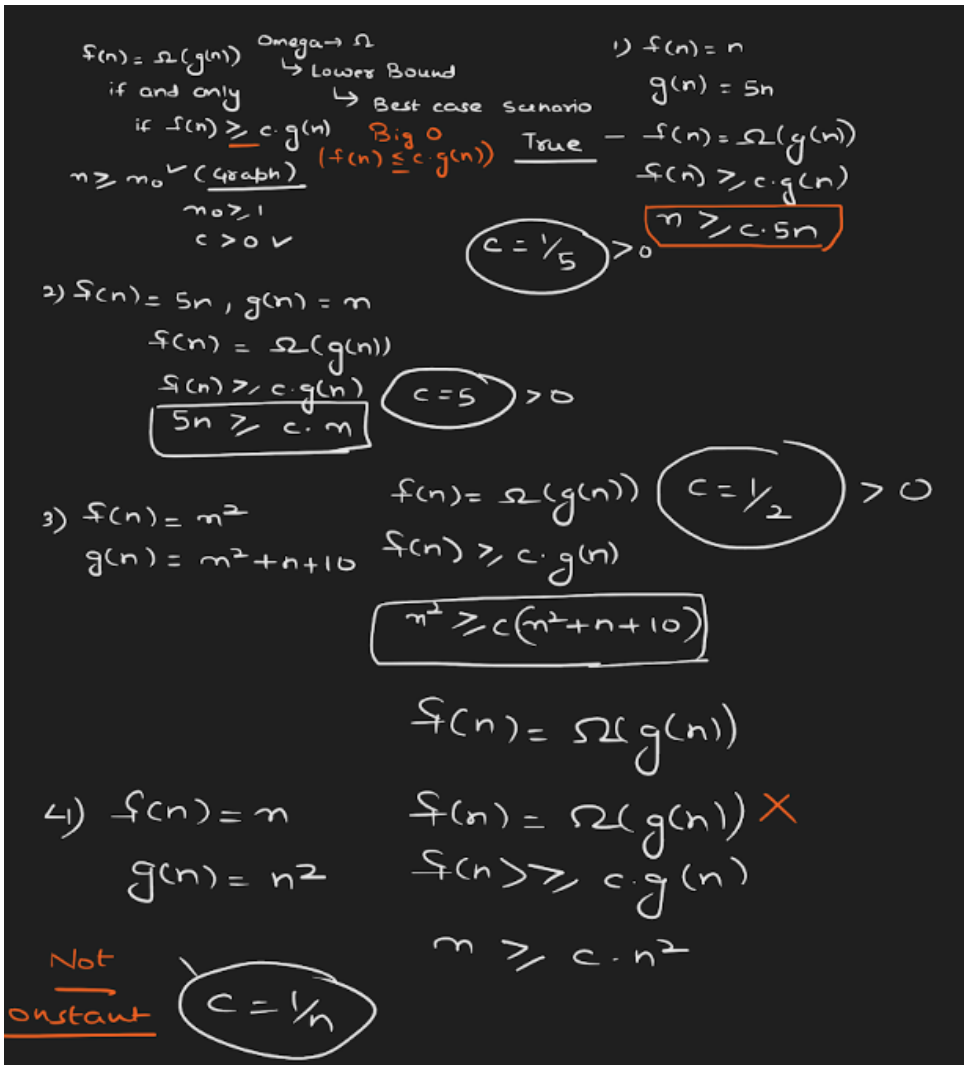
- Big O notation
- Omega Notation
- Theta Notation

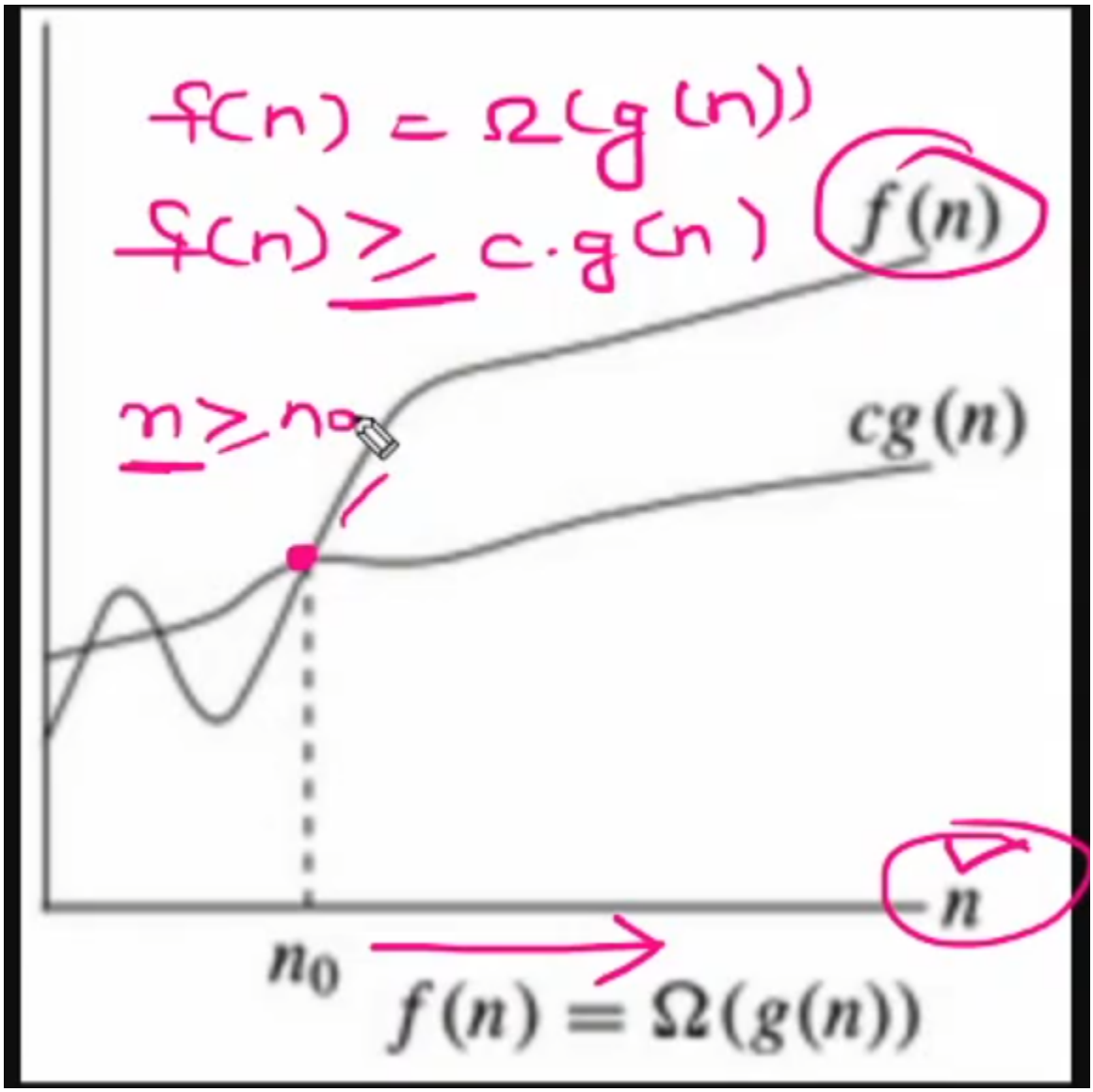
1. Big O notation: WORST CASE SCENARIO(upper bound)



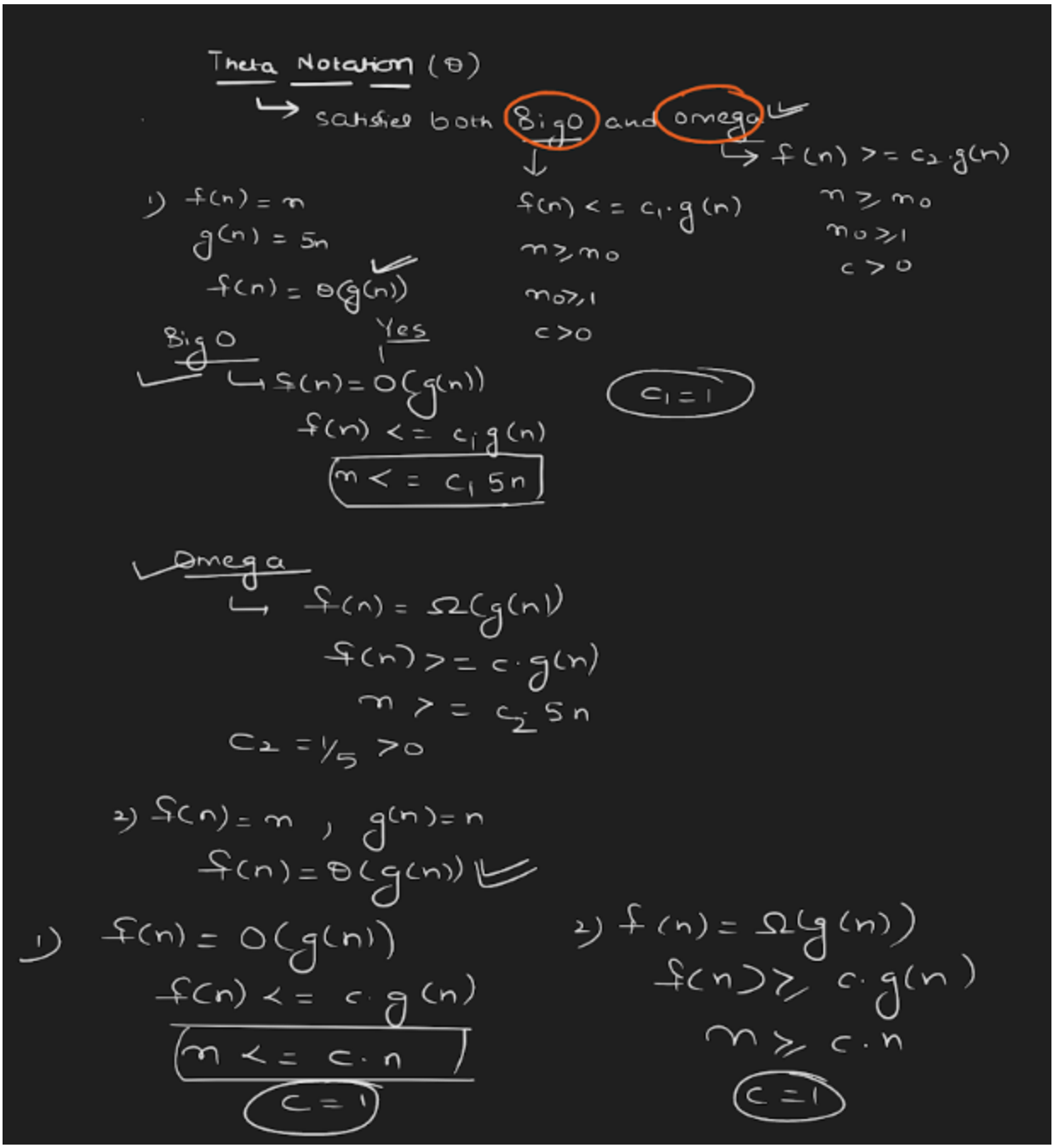


2. Omega Notation: Best Case Scenario(Lower Bound)


f(n) = Ω(g(n)) Omega → Ω
if and only ↳ Lower Bound
if f(n) ≥ c·g(n) ↳ Best case scenario
n ≥ n₀ (graph) Big O
n₀ ≥ 1 (f(n) ≤ c·g(n))
c > 0
1) f(n) = n
g(n) = 5n
True
f(n) = Ω(g(n))
f(n) ≥ c·g(n)
n ≥ c·5n
c = 1/5 > 0
2) f(n) = 5n, g(n) = n
f(n) = Ω(g(n))
f(n) ≥ c·g(n)
5n ≥ c·n
c = 5 > 0
3) f(n) = n²
g(n) = n²+n+10
f(n) = Ω(g(n))
f(n) ≥ c·g(n)
n² ≥ c(n²+n+10)
c = 1/2 > 0
f(n) = Ω(g(n))
4) f(n) = n
g(n) = n²
f(n) = Ω(g(n)) ✗
f(n) ≥ c·g(n)
n ≥ c·n²
Not constant
c = 1/n



3. Theta Notation: Satisfies both, omega and theta notation


Theta Notation (Θ)
satisfied both Big O and omega
f(n) >= c2 . g(n)
n ≥ n0
n0 ≥ 1
c > 0
1) f(n) = n
g(n) = 5n
f(n) = Θ(g(n))
f(n) <= c1 . g(n)
n ≥ n0
n0 ≥ 1
c > 0
Big O
Yes
f(n) = O(g(n))
c1 = 1
f(n) <= c1 g(n)
n <= c1 5n
Omega
f(n) = Ω(g(n))
f(n) >= c . g(n)
n >= c2 5n
c2 = 1/5 > 0
2) f(n) = n , g(n) = n
f(n) = Θ(g(n))
1) f(n) = O(g(n))
f(n) <= c . g(n)
n <= c . n
c = 1
2) f(n) = Ω(g(n))
f(n) ≥ c . g(n)
n ≥ c . n
c = 1