

Asymptotic Notations: Big O, Big Omega and Big Theta Explained (With Notes)

CodeWithHarry

We'll talk a little bit about asymptotic notation. We talked about order. We talked about ordering. We have primarily 3 types of asymptotic notation big O, big Theta (Θ) and big Omega (Ω). Big O is represented by capital (O), which is in our English. Big O is set to be $O(g(n))$ if and only if there exist a constant (c) and a constant n -node such that $0 \leq f(n) \leq cg(n)$ is $O(g(N))$. If you watch this video completely then I guarantee that you will understand these three notations. Mathematically, mathematically this function can be anything. When we do analysis of algorithms comparing any 2 algorithms then $f(n)$ will be time and what is n , it's input ok, size of input. $G(n)$ is your function which will come inside the big O. $O(n^2)$ is Anything Can Be Algorithm it is $g(n)$ that will be here and which is your algorithm. If you guys can find any such constant (C) and (n)-node, then $f(n)$ is $O(g(n))$. This is the mathematical definition of big O. If you can't find it then it's not $f(n)$ is O . This question is its own truth, it has validity, it will remain valid.

Preview from Notesale.co.uk
Page 1 of 1