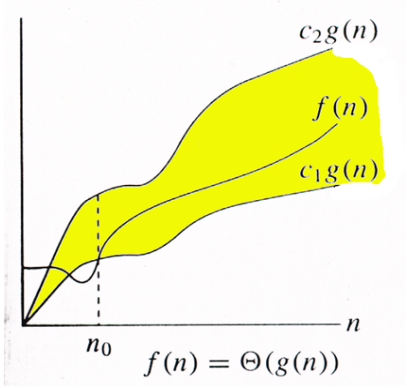
Analysis of Algorithms | Set 3 (Asymptotic Notations)

We have discussed Asymptotic Analysis, and Worst, Average and Best Cases of Algorithms. The main idea of asymptotic analysis is to have a measure of efficiency of algorithms that doesn't depend on machine specific constants, and doesn't require algorithms to be implemented and time taken by programs to be compared. Asymptotic notations are mathematical tools to represent time complexity of algorithms for asymptotic analysis. The following 3 asymptotic notations are mostly used to represent time complexity of algorithms.

1) Θ Notation: The theta notation bounds a functions from above and below, so it defines exact asymptotic behavior.
A simple way to get Theta notation of an expression is to drop low order terms and ignore leading constants. For example, consider the following expression.
$3n^3 + 6n^2 + 6000 = \Theta(n^3)$
Dropping lower order terms is always fine because there will always be a n0 after which $\Theta(n^3)$ beats $\Theta(n^2)$ irrespective of the constants involved.



For a given function g(n), we denote Θ(g(n)) is following set of functions.

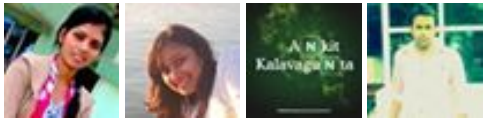
```
Θ((g(n)) = {f(n): there exist positive constants c1, c2 and n0 such that
                  0 <= c1*g(n) <= f(n) <= c2*g(n) for all n >= n0}
```

The above definition means, if f(n) is theta of g(n), then the value f(n) is always between c1*g(n) and c2*g(n) for large values of n (n >= n0). The definition of theta also requires that f(n) must be non-negative for values of n greater than n0.

Google™ Custom Search

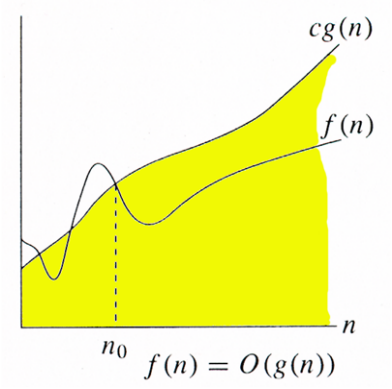
GeeksforGeeks

53,522 people like GeeksforGeeks.



open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

2) Big O Notation: The Big O notation defines an upper bound of an algorithm, it bounds a function only from above. For example, consider the case of Insertion Sort. It takes linear time in best case and quadratic time in worst case. We can safely say that the time complexity of Insertion sort is O(n^2). Note that O(n^2) also covers linear time.
If we use Θ notation to represent time complexity of Insertion sort, we have to use two statements for best and worst cases:
1. The worst case time complexity of Insertion Sort is Θ(n^2).
2. The best case time complexity of Insertion Sort is Θ(n).

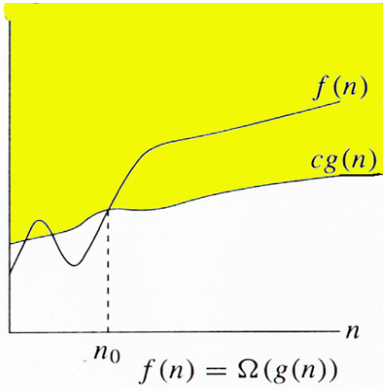


The Big O notation is useful when we only have upper bound on time complexity of an algorithm. Many times we easily find an upper bound by simply looking at the algorithm.

```
O(g(n)) = { f(n): there exist positive constants c and n0 such that
            0 <= f(n) <= cg(n) for all n >= n0}
```

3) Ω Notation: Just as Big O notation provides an asymptotic upper bound on a function, Ω notation provides an asymptotic lower bound.

Ω Notation< can be useful when we have lower bound on time complexity of an algorithm. As discussed in the previous post, the best case performance of an algorithm is generally not useful, the Omega notation is the least used notation among all three.



For a given function g(n), we denote by Ω(g(n)) the set of functions.

```
Ω (g(n)) = {f(n): there exist positive constants c and n0 such that
                          0 <= cg(n) <= f(n) for all n >= n0}.
```

Let us consider the same Insertion sort example here. The time complexity of Insertion Sort can be written as Ω(n), but it is not a very useful information about insertion sort, as we are generally interested in worst case and sometimes in average case.

Geometric Algorithms

Popular Posts

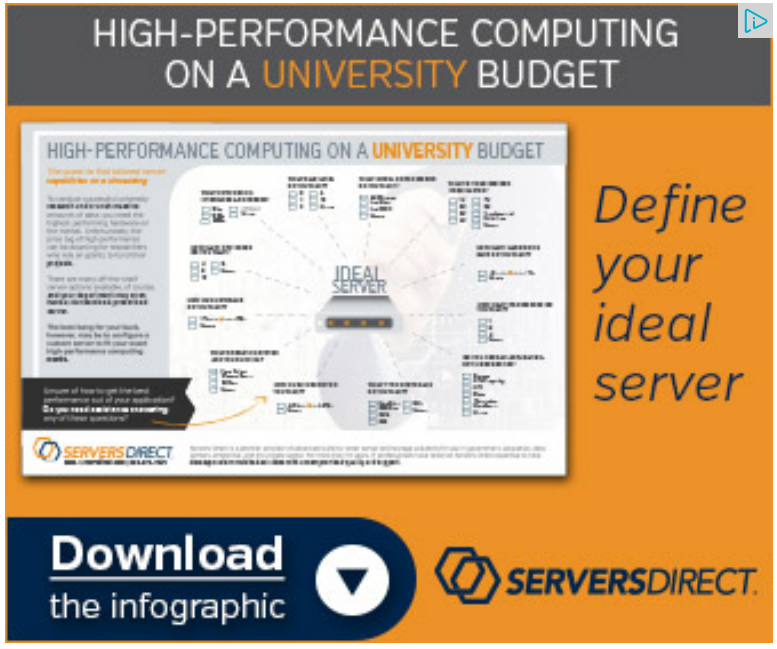
open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Exercise:
Which of the following statements is/are valid?
1. Time Complexity of QuickSort is Θ(n^2)
2. Time Complexity of QuickSort is O(n^2)
3. For any two functions f(n) and g(n), we have f(n) = Θ (g(n)) if and only if f(n) = O(g(n)) and f(n) = Ω(g(n)).
4. Time complexity of all computer algorithms can be written as Ω (1)

References:
Lec 1 | MIT (Introduction to Algorithms)

Introduction to Algorithms 3rd Edition by Clifford Stein, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest

This article is contributed by **Abhay Rathi**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Related Tpoics:

- Analysis of Algorithms | Set 4 (Analysis of Loops)

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

- NP-Completeness | Set 1 (Introduction)
- Static and Dynamic Libraries | Set 1
- The Ubiquitous Binary Search | Set 1
- Reservoir Sampling
- Analysis of Algorithms | Set 2 (Worst, Average and Best Cases)
- Analysis of Algorithms | Set 1 (Asymptotic Analysis)
- Scope rules in C

27 Tweet 4 1

Writing code in comment? Please use **ideone.com** and share the link here.

11 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Marsha Donna • 4 months ago

so does this mean that
big O notation corresponds to worst case time complexity,
big omega notation corresponds to best case time complexity,

big theta notation corresponds to average case time complexity???

9 ▲ ▼ •

darubramha ➦ Marsha Donna • a month ago

Somewhat correct.

big O notation corresponds to worst case time complexity, generally it
T(n) [time complexity] = O(n) then it means, T(n) can have maximum

big omega notation corresponds to best case complexity, it is similar t
case.

705 Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...
Write a C program to Delete a Tree. · 29 minutes ago

kzs please provide solution for the problem...
Backtracking | Set 2 (Rat in a Maze) · 32 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...
Root to leaf path sum equal to a given number · 57 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...
Count trailing zeroes in factorial of a number · 59 minutes ago

newCoder3006 If the array contains negative numbers also. We...
Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...
Find subarray with given sum · 1 hour ago

AdChoices
▶ Algorithm Design
▶ C Algorithm
▶ Notations

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

But, big theta notation, tells us that the time complexity of the algorithm given function, which means if T(n) = theta(n), the time complexity can be better than c2n, where c1,c2 are constants. It does not mean the av towards "=", i.e c1n <= T(n) <= c2n

1 ^ | v •

Ankur Teotia ➚ Marsha Donna • 2 months ago

i have the same doubt , someone please clarify.
it seems to be the case but no where i see this explicitly mentioned.

1 ^ | v •

noob • 7 months ago

1. Time Complexity of QuickSort is (n^2) -----> NO
2. Time Complexity of QuickSort is O(n^2) -----> YES
3. For any two functions f(n) and g(n), we have f(n) = (g(n)) if and only if f(n) =
4. Time complexity of all computer algorithms can be written as (1) -----> YES

8 ^ | v •

Chaithanya Kanumolu ➚ noob • 4 months ago

I am confused with 4th point. Since best case for insertion sort is Theta write it as Theta (1)?

1 ^ | v •

ibn ➚ noob • 5 months ago

Why was question 4 Yes?

^ | v •

Utkarsh Gupta ➚ ibn • 3 months ago

If f(n) = O(n) then it means f(n) can be less than or equal to ord
If f(n) = Θ(n) then it means f(n) is equal to the order of n.
If f(n) = Ω(n) then it means f(n) is greater than or equal to order

AdChoices
▶ Log Analysis
▶ It Analysis
▶ DHA Omega 3
AdChoices
▶ Computer Geeks
▶ Linear Function
▶ N Analysis

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

So if you see Ω(1), it means that this is the smallest lower bour
algorithm f(n) = Ω(1). But it can be even more precise that is f(r
different conditions.

mog ibn • 4 months ago

bcoz oemga always provide a lower bound on running time of a
would be omega(1),it cant be less than omega(1).....so we can
as omega(1).

Sourabh mog • 3 months ago

For a sorting algorithm, best case can not be better thai
elements before saying that it is sorted, so I guess ansv

1

rohit Sourabh • a month ago

1<n ==""> yes

Sourabh rohit • 17 days ago

Well you got me there, I saw this answer again today ar
IMO omega(1) is true for any algorithm (its a lower bour

Subscribe

Add Disqus to your site

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com