

# GeeksforGeeks

A computer science portal for geeks

## GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

## Analysis of Algorithms | Set 1 (Asymptotic Analysis)

### *Why performance analysis?*

There are many important things that should be taken care of, like user friendliness, modularity, security, maintainability, etc. Why to worry about performance?

The answer to this is simple, we can have all the above things only if we have performance. So performance is like currency through which we can buy all the above things. Another reason for studying performance is – speed is fun!

### *Given two algorithms for a task, how do we find out which one is better?*

One naive way of doing this is – implement both the algorithms and run the two programs on your computer for different inputs and see which one takes less time. There are many problems with this approach for analysis of algorithms.

- 1) It might be possible that for some inputs, first algorithm performs better than the second. And for some inputs second performs better.
- 2) It might also be possible that for some inputs, first algorithm perform better on one machine and the second works better on other machine for some other inputs.

[Asymptotic Analysis](#) is the big idea that handles above issues in analyzing algorithms. In Asymptotic Analysis, we evaluate the performance of an algorithm in terms of input size (we don't measure the actual running time). We calculate, how does the time (or space) taken by an algorithm increases with the input size.

For example, let us consider the search problem (searching a given item) in a sorted array. One way to search is Linear Search (order of growth is linear) and other way is Binary Search (order of growth is logarithmic). To understand how Asymptotic Analysis solves the above mentioned problems in analyzing algorithms, let us say we run the Linear Search on a fast computer and Binary Search on a slow computer. For small values of input array size  $n$ , the fast computer may take less time. But, after certain value of input array size, the Binary Search will definitely start taking less time compared to the Linear Search even though the Binary Search is being run on a slow machine. The reason is the order of growth of Binary Search with respect to input size logarithmic while the order of growth of Linear Search is linear. So the machine dependent constants can always be ignored after certain values of input size.

### ***Does Asymptotic Analysis always work?***

Asymptotic Analysis is not perfect, but that's the best way available for analyzing algorithms. For example, say there are two sorting algorithms that take  $1000n\log n$  and  $2n\log n$  time respectively on a machine. Both of these algorithms are asymptotically same (order of growth is  $n\log n$ ). So, With Asymptotic Analysis, we can't judge which one is better as we ignore constants in Asymptotic Analysis. Also, in Asymptotic analysis, we always talk about input sizes larger than a constant value. It might be possible that those large inputs are never given to your software and an algorithm which is asymptotically slower, always performs better for your particular situation. So, you may end up choosing an algorithm that is Asymptotically slower but faster for your software.

We will covering more on analysis of algorithms in some more posts on this topic.

### **References:**

[MIT's Video lecture 1 on Introduction to Algorithms](#) .

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

### **Related Topics:**

- [An interesting time complexity question](#)
- [A Problem in Many Binary Search Implementations](#)
- [Analysis of Algorithms | Set 4 \(Analysis of Loops\)](#)
- [Analysis of Algorithms | Set 3 \(Asymptotic Notations\)](#)
- [NP-Completeness | Set 1 \(Introduction\)](#)
- [Static and Dynamic Libraries | Set 1](#)
- [The Ubiquitous Binary Search | Set 1](#)
- [Reservoir Sampling](#)

Like { 15 }

Tweet { 1 }

g+1 { 4 }

Writing code in comment? Please use [ideone.com](http://ideone.com) and share the link here.

16 Comments

GeeksforGeeks

1 Login ▾

♥ Recommend

🔗 Share

Sort by Newest ▾



Join the discussion...

**mADAN kapoor** • 6 months ago

data graphs as pictures for showing would improve the asymptotic Analysis definition

1 ^ | ▾ • Reply • Share ›

**aRUN** • a year ago

PLZ TELL ME THE TIME COMPLEXITY OF FOLLOWING PROGRAM IT'S KIND URGENT

#include &lt;c:\opencv\build\include\opencv\cvaux.h&gt;

#include &lt;c:\opencv\build\include\opencv\highgui.h&gt;

#include &lt;c:\opencv\build\include\opencv\cxcore.h&gt;

#include &lt;omp.h&gt;

#include &lt;time.h&gt;

#include &lt;iostream&gt;

#include &lt;fstream&gt;

#include &lt;cmath&gt;

#include &lt;stdio.h&gt;

#include &lt;stdlib.h&gt;

[see more](#)

30 ^ | ▾ • Reply • Share ›

**Jovaughn** • a year ago

Can you clarify the statement:

"we evaluate the performance of an algorithm in terms of input size (we don't measure the actual running time)." Can you clarify what running time is? Running time in other books mean steps or operations executed.

^ | ▾ • Reply • Share ›

**narsna** ↗ jovaugnn · a year ago

Running time is time taken by program to run and display desired output . Asymptotic Analysis is used to pick up which algorithm is better. Asymptotic notation doesn't give actual running time it will says how running time is increased with respect to size of input .

4 ^ | v · Reply · Share ›

**Lily** · a year ago

very helpful website..

^ | v · Reply · Share ›

**hari** · a year ago

useful

^ | v · Reply · Share ›

**Rajavenu Kyatham** · 2 years ago

awesome website....

^ | v · Reply · Share ›

**Kush Pandey** · 2 years ago

good job

^ | v · Reply · Share ›

**hxgxs1** · 2 years ago

Doubt!!!

It says that

"in Asymptotic analysis, we always talk about input sizes larger than a constant value."

In the case of  $1000n\log n$  and  $2n\log n$  the constant values are 1000 and 2 right??

14 ^ | v · Reply · Share ›

**Robin Thomas** ↗ hxgxs1 · 2 years ago

Nope. 1000 and 2 are merely machine dependent constants. Those are ignored in asymptotic analysis.

In asymptotic analysis, we always assume the input size  $n \geq n_0$ , where  $n_0$  is the constant. If you check the formal definition of big O, or big Omega or the other classes, you shall understand it.

12 ^ | v · Reply · Share ›

**hari** ↗ Robin Thomas · a year ago

you are right man

^ | v · Reply · Share ›

**Sampat S Magi** · 2 years ago



great site. knowledgeable.

^ | v • Reply • Share ›



**Deepak Singh** • 2 years ago

great work

^ | v • Reply • Share ›



**Ash** • 2 years ago

Didn't you mean "For small values of input array size  $n$ , the \*fast\* computer may take less time"?

^ | v • Reply • Share ›



**GeeksforGeeks** → Ash • 2 years ago

Thanks for pointing this out. There was a typo. We have corrected the statement.

5 ^ | v • Reply • Share ›



**latha** • 3 years ago

nice one thank u:)

^ | v • Reply • Share ›



Subscribe



Add Disqus to your site



Privacy



**GeeksforGeeks**

Like

97,207 people like [GeeksforGeeks](#).



Facebook social plugin

- [Interview Experiences](#)
- [Advanced Data Structures](#)

- [Dynamic Programming](#)
- [Greedy Algorithms](#)
- [Backtracking](#)
- [Pattern Searching](#)
- [Divide & Conquer](#)
- [Mathematical Algorithms](#)
- [Recursion](#)
- [Geometric Algorithms](#)

## • Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

## • Recent Comments

- [It\\_k](#)

i need help for coding this function in java...

[Java Programming Language](#) · [1 hour ago](#)

- [Piyush](#)

What is the purpose of else if (recStack[\*i])...

[Detect Cycle in a Directed Graph](#) · [1 hour ago](#)

- [Andy Toh](#)

My compile-time solution, which agrees with the...

[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [1 hour ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [1 hour ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [2 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [2 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team