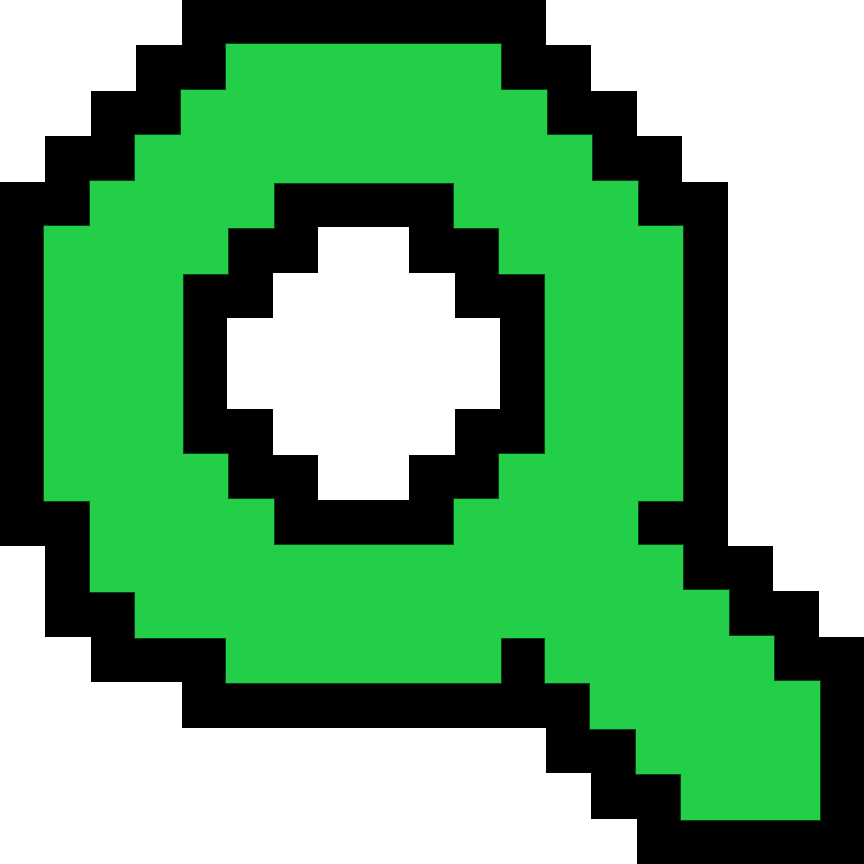
1

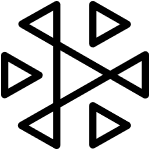
https://hackernoon.com/what-does-the-time-complexity-o-log-n-actually-mean-45f94bb5bfbf





[Start Writing](https://app.hackernoon.com/new)Log in

* [Read](https://hackernoon.com/what-does-the-time-complexity-o-log-n-actually-mean-45f94bb5bfbf)
* [Top Stories](https://hackernoon.com/tagged/hackernoon-top-story)
* [Write](https://hackernoon.com/what-does-the-time-complexity-o-log-n-actually-mean-45f94bb5bfbf" \l "sponsor)
* [Learn](https://bit.ly/3rG0tZy)
* [STARTUPS](https://startups.hackernoon.com/)
* [Tech Giants](https://hackernoon.com/companies)
* [About](https://hackernoon.com/what-does-the-time-complexity-o-log-n-actually-mean-45f94bb5bfbf)
* [Help](https://hackernoon.com/what-does-the-time-complexity-o-log-n-actually-mean-45f94bb5bfbf)
* [Sponsor](https://sponsor.hackernoon.com/)

[Build with Interactive Live Video](https://ivs.rocks/?ref=hackernoon)

What does the time complexity O(log n) actually mean? by[@maazrk](https://hackernoon.com/u/maazrk)

# What does the time complexity O(log n) actually mean?

Originally published by Maaz on [May 28th 2017](https://hackernoon.com/archives/2017/05/28) 224,943 reads









O(1) means an operation is done to reach an element directly (like a dictionary or hash table) O(n) means first we would have to search it by checking n elements, but what could O(log n) possibly mean? Binary search algorithm is one of the most complex algorithms in computer science. The log n actually means something doesn’t it? A type of behavior nothing else can represent. When working in the field of computer science, it is always helpful to know such stuff.

### [@maazrk](https://hackernoon.com/u/maazrk)

### Maaz

Knowing the complexity of algorithms beforehand is one thing, and other thing is knowing the reason behind it being like that.

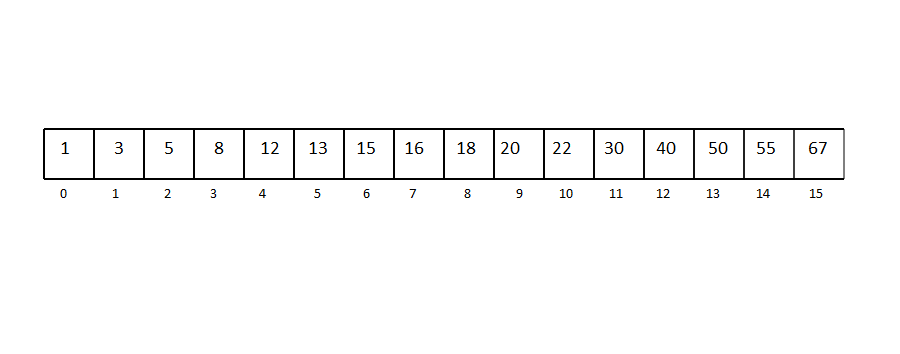
Whether you’re a CS graduate or someone who wants to deal with optimization problems effectively, this is something that you must understand if you want to use your knowledge for solving actual problems.

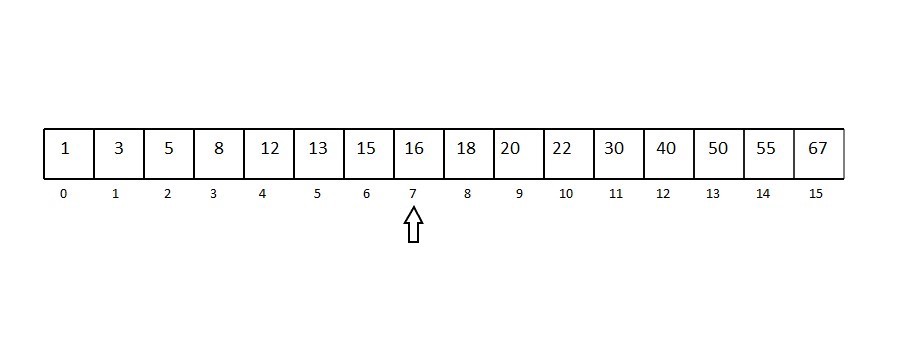
Complexities like O(1) and O(n) are simple and straightforward. O(1) means an operation which is done to reach an element directly (like a dictionary or hash table), O(n) means first we would have to search it by checking n elements, but what could O(log n) possibly mean?

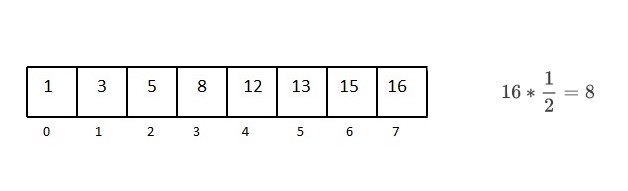
One place where you might have heard about O(log n) time complexity the first time is Binary search algorithm. So there must be some type of behavior that algorithm is showing to be given a complexity of log n. Let us see how it works.

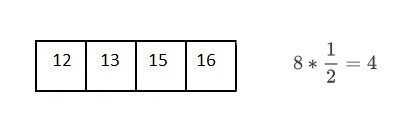
Since binary search has a best case efficiency of O(1) and worst case (average case) efficiency of O(log n), we will look at an example of the worst case. Consider a sorted array of 16 elements.

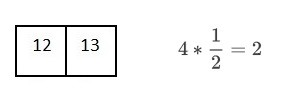
For the worst case, let us say we want to search for the the number 13.

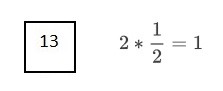








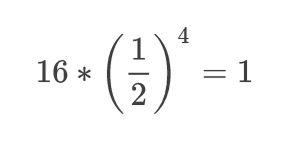




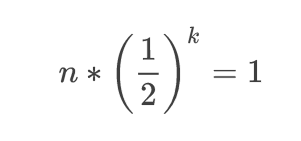
You can see that after every comparison with the middle term, our searching range gets divided into half of the current range.

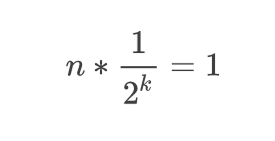
So, for reaching one element from a set of 16 elements, we had to divide the array 4 times,

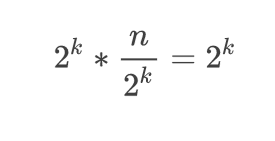
We can say that,

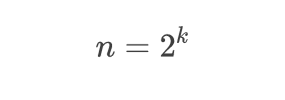


Similarly, for n elements,





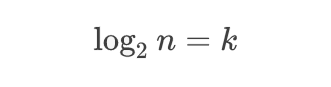




Now, let us look at the definition of logarithm, it says that

A quantity representing the power to which a fixed number (the base) must be raised to produce a given number.

Which makes our equation into



So the log n actually means something doesn’t it? A type of behavior nothing else can represent.

Well, i hope the idea of it is clear in your mind. When working in the field of computer science, it is always helpful to know such stuff (and is quite interesting too). Who knows, maybe you’re the one in your team who is able to find an optimized solution for a problem, just because you know what you’re dealing with. Good luck!