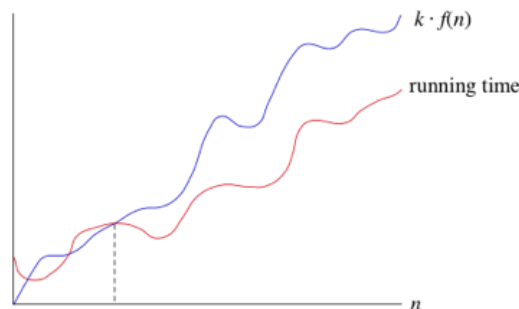


## Big-O notation

We use big- $\Theta$  notation to asymptotically bound the growth of a running time to within constant factors above and below. Sometimes we want to bound from only above. For example, although the worst-case running time of binary search is  $\Theta(\lg n)$ , it would be incorrect to say that binary search runs in  $\Theta(\lg n)$  time in *all* cases. What if we find the target value upon the first guess? Then it runs in  $\Theta(1)$  time. The running time of binary search is never worse than  $\Theta(\lg n)$ , but it's sometimes better. It would be convenient to have a form of asymptotic notation that means "the running time grows at most this much, but it could grow more slowly." We use "big-O" notation for just such occasions.

If a running time is  $O(f(n))$ , then for large enough  $n$ , the running time is at most  $k \cdot f(n)$  for some constant  $k$ . Here's how to think of a running time that is  $O(f(n))$ :



We say that the running time is "big-O of  $f(n)$ " or just "O of  $f(n)$ ." We use big-O notation for **asymptotic upper bounds**, since it bounds the growth of the running time from above for large enough input sizes.

Now we have a way to characterize the running time of binary search in all cases. We can say that the running time of binary search is *always*  $O(\lg n)$ . We can make a stronger statement about the worst-case running time: it's  $\Theta(\lg n)$ . But for a blanket statement that covers all cases, the strongest statement we can make is that binary search runs in  $O(\lg n)$  time.

If you go back to the definition of big- $\Theta$  notation, you'll notice that it looks a lot like big-O notation, except that big- $\Theta$  notation bounds the running from both above and below, rather than just from above. If we say that a running time is  $\Theta(f(n))$  in a particular situation, then it's also  $O(f(n))$ . For example, we can say that because the worst-case running time of binary search is  $\Theta(\lg n)$ , it's also  $O(\lg n)$ . The converse is not necessarily true: as we've seen, we can say that binary search always runs in  $O(\lg n)$  time but *not* that it always runs in  $\Theta(\lg n)$  time.

Because big-O notation gives only an asymptotic upper bound, and not an asymptotically tight bound, we can make statements that at first blush seem incorrect, but are technically correct. For example, it is absolutely correct to say that binary search runs in  $O(n)$  time.

That's because the running time grows no faster than a constant times  $n$ . In fact, it grows slower. Think of it this way. Suppose you have 10 dollars in your pocket. You go up to your friend and say, "I have an amount of money in my pocket, and I guarantee that it's no more than one million dollars." Your statement is absolutely true, though not terribly precise. One million dollars is an upper bound on 10 dollars, just as  $O(n)$  is an upper bound on the running time of binary search. Other, imprecise, upper bounds on binary search would be  $O(n^2)$ ,  $O(n^3)$ , and  $O(2^n)$ . But none of  $\Theta(n)$ ,  $\Theta(n^2)$ ,  $\Theta(n^3)$ , and  $\Theta(2^n)$  would be correct to describe the running time of binary search in any case.

This content is a collaboration of [Dartmouth Computer Science](#) professors [Thomas Cormen](#) and [Devin Balkcom](#), plus the Khan Academy computing curriculum team. The content is licensed [CC-BY-NC-SA](#).

◀ ALGORITHMS

Asymptotic notation

Asymptotic notation

Big-Θ (Big-Theta) notation

Functions in asymptotic notation

Quiz: Comparing function growth

Big-O notation

Big-Ω (Big-Omega) notation

Quiz: Asymptotic notation

NEXT SECTION:

Selection sort

Questions

Tips & Thanks

Top

Recent

Report a mistake in the article

Ask a question...

Example:  
The link in the third paragraph is broken.  

Report a mistake in the article

Discuss the site  
For general discussions about Khan Academy, [click here](#).

Flag inappropriate posts  
Here are posts to avoid making. If you do encounter them, flag them for attention from our Guardians.  
abuse

- disrespectful or offensive
- an advertisement

not helpful

- low quality
- not about the video topic
- soliciting votes or seeking badges
- a homework question
- a duplicate answer
- repeatedly making the same post

wrong category

- a tip or thanks in Questions
- a question in Tips & Thanks
- an answer that should be its own question

about the site

- a question about Khan Academy  
[\(Visit our FAQ\)](#)
- a post about badges
- a technical problem with the site  
[\(Report a problem\)](#)
- a request for [features](#)

How does Big-Theta not describe all cases of running time? Both Big-O and Big-Theta are limited to higher values of n, and they share the upper bound if I'm understanding correctly. How then does adding a lower bound become less precise, or in other words what makes adding a lower bound a less comprehensive description that does not cover all cases like Big-O does?

Edit: so reading the Big-Omega section I gather that Big-Theta does not refer the range between Big-O and Big-Omega. It seems... [\(more\)](#)

4 Votes ▲ ▼ • [Comment](#) • [Flag](#) 2 months ago by jandattippett

Big-Theta puts an upper and lower bound on all cases for  $n >$  some constant  $k$ . It describes the *rate* at which the time it takes for an algorithm to run will increase as  $n$  increases.

3 Votes ▲ ▼ • [Comment](#) • [Flag](#) 2 months ago by John Sutton

[Show all 4 answers](#) • [Answer this question](#)

If I'm not mistaken, the first paragraph is a bit misleading.

Before, we used big-Theta notation to describe the *worst case* running time of binary search, which is  $\Theta(\lg n)$ . The *best case* running time is a completely different matter, and it is  $\Theta(1)$ .

That is, there are (at least) three different types of running times that we generally consider: best case, average/expected case, and worst case. Usually it's the latter two that are the most useful. For binary search, the best case time is  $\Theta(1)$ , and... [\(more\)](#)

3 Votes ▲ ▼ • [Comment](#) • [Flag](#) 2 months ago by William Chargin

I would generally agree with this. I would also state that when the type of running time is not stated it is generally assumed that the type is worst case.

2 Votes ▲ ▼ • [1 Comment](#) • [Flag](#) 2 months ago by Cameron

Answer this question...

ABOUT

Our Mission

You Can Learn Anything

Our Team

Our Interns

SUPPORT

Help center

CONTACT US

Contact

COACHING

Coach Reports

Coach Resources

Case Studies

Common Core

CAREERS

Full Time

Internships

CONTRIBUTE

INTERNATIONAL

Change language

Translate our content

SOCIAL

https://www.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/big-o-notation

2/3

|                                         |                       |                                |                            |
|-----------------------------------------|-----------------------|--------------------------------|----------------------------|
| <a href="#">Our Content Specialists</a> | <a href="#">Press</a> | <a href="#">Donate</a>         | <a href="#">Facebook</a>   |
| <a href="#">Our Board</a>               |                       | <a href="#">Volunteer</a>      | <a href="#">Twitter</a>    |
|                                         |                       | <a href="#">Our Supporters</a> | <a href="#">Blog</a>       |
|                                         |                       | <a href="#">Shop KA</a>        | <a href="#">Life at KA</a> |
|                                         |                       |                                | <input type="text"/>       |