

Big-O Cheat Sheet

[Download PDF](#)

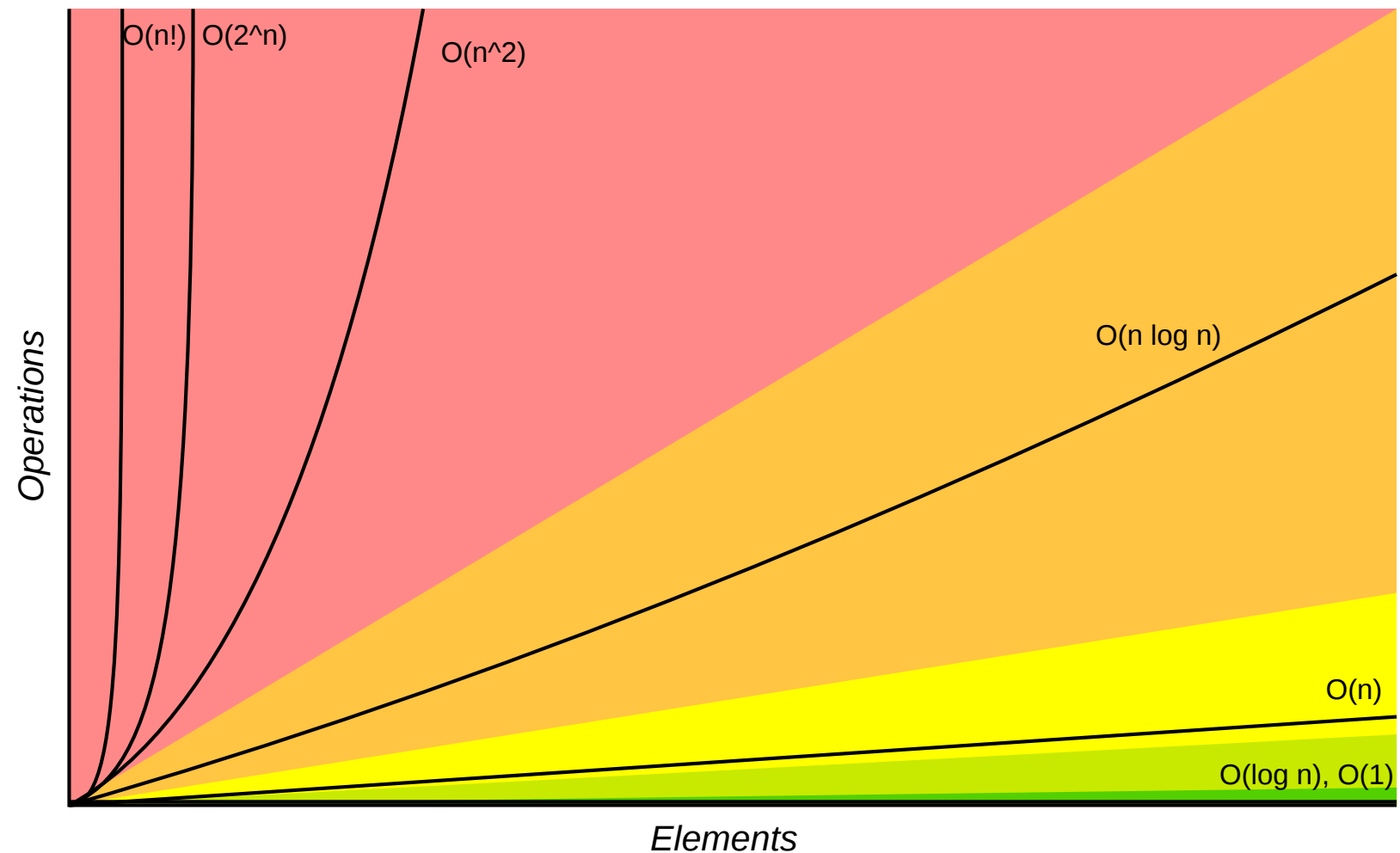
Know Thy Complexities!

Hi there! This webpage covers the space and time Big-O complexities of common algorithms used in Computer Science. When preparing for technical interviews in the past, I found myself spending hours crawling the internet putting together the best, average, and worst case complexities for search and sorting algorithms so that I wouldn't be stumped when asked about them. Over the last few years, I've interviewed at several Silicon Valley startups, and also some bigger companies, like Google, Facebook, Yahoo, LinkedIn, and eBay, and each time that I prepared for an interview, I thought to myself "Why hasn't someone created a nice Big-O cheat sheet?". So, to save all of you fine folks a ton of time, I went ahead and created one. Enjoy! - [Eric](#)

If you're trying to catch them all, you might also check out the [Pokemon Go Evolution Chart](#).

Big-O Complexity Chart

| | | | | |
|----------|-----|------|------|-----------|
| Horrible | Bad | Fair | Good | Excellent |
|----------|-----|------|------|-----------|



Common Data Structure Operations

| Data Structure | Time Complexity | | | | | | | | Space Complexity |
|----------------|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------------|
| | Average | | | | Worst | | | | Worst |
| | Access | Search | Insertion | Deletion | Access | Search | Insertion | Deletion | |
| <u>Array</u> | $\theta(1)$ | $\theta(n)$ | $\theta(n)$ | $\theta(n)$ | $\theta(1)$ | $\theta(n)$ | $\theta(n)$ | $\theta(n)$ | $\theta(n)$ |

| | | | | | | | | | |
|---------------------------|-------------------|-------------------|-------------------|-------------------|--------------|--------------|--------------|--------------|----------------|
| <u>Stack</u> | $\Theta(n)$ | $\Theta(n)$ | $\Theta(1)$ | $\Theta(1)$ | $O(n)$ | $O(n)$ | $O(1)$ | $O(1)$ | $O(n)$ |
| <u>Queue</u> | $\Theta(n)$ | $\Theta(n)$ | $\Theta(1)$ | $\Theta(1)$ | $O(n)$ | $O(n)$ | $O(1)$ | $O(1)$ | $O(n)$ |
| <u>Singly-Linked List</u> | $\Theta(n)$ | $\Theta(n)$ | $\Theta(1)$ | $\Theta(1)$ | $O(n)$ | $O(n)$ | $O(1)$ | $O(1)$ | $O(n)$ |
| <u>Doubly-Linked List</u> | $\Theta(n)$ | $\Theta(n)$ | $\Theta(1)$ | $\Theta(1)$ | $O(n)$ | $O(n)$ | $O(1)$ | $O(1)$ | $O(n)$ |
| <u>Skip List</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n \log(n))$ |
| <u>Hash Table</u> | N/A | $\Theta(1)$ | $\Theta(1)$ | $\Theta(1)$ | N/A | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |
| <u>Binary Search Tree</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |
| <u>Cartesian Tree</u> | N/A | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | N/A | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |
| <u>B-Tree</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(n)$ |
| <u>Red-Black Tree</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(n)$ |
| <u>Splay Tree</u> | N/A | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | N/A | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(n)$ |
| <u>AVL Tree</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(\log(n))$ | $O(n)$ |
| <u>KD Tree</u> | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $\Theta(\log(n))$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |

Array Sorting Algorithms

| Algorithm | Time Complexity | | | Space Complexity |
|-----------------------|---------------------|------------------------|-------------------|------------------|
| | Best | Average | Worst | Worst |
| <u>Quicksort</u> | $\Omega(n \log(n))$ | $\Theta(n \log(n))$ | $O(n^2)$ | $O(\log(n))$ |
| <u>Mergesort</u> | $\Omega(n \log(n))$ | $\Theta(n \log(n))$ | $O(n \log(n))$ | $O(n)$ |
| <u>Timsort</u> | $\Omega(n)$ | $\Theta(n \log(n))$ | $O(n \log(n))$ | $O(n)$ |
| <u>Heapsort</u> | $\Omega(n \log(n))$ | $\Theta(n \log(n))$ | $O(n \log(n))$ | $O(1)$ |
| <u>Bubble Sort</u> | $\Omega(n)$ | $\Theta(n^2)$ | $O(n^2)$ | $O(1)$ |
| <u>Insertion Sort</u> | $\Omega(n)$ | $\Theta(n^2)$ | $O(n^2)$ | $O(1)$ |
| <u>Selection Sort</u> | $\Omega(n^2)$ | $\Theta(n^2)$ | $O(n^2)$ | $O(1)$ |
| <u>Tree Sort</u> | $\Omega(n \log(n))$ | $\Theta(n \log(n))$ | $O(n^2)$ | $O(n)$ |
| <u>Shell Sort</u> | $\Omega(n \log(n))$ | $\Theta(n(\log(n))^2)$ | $O(n(\log(n))^2)$ | $O(1)$ |

| | | | | |
|----------------------|---------------|---------------------|----------------|----------|
| <u>Bucket Sort</u> | $\Omega(n+k)$ | $\Theta(n+k)$ | $O(n^2)$ | $O(n)$ |
| <u>Radix Sort</u> | $\Omega(nk)$ | $\Theta(nk)$ | $O(nk)$ | $O(n+k)$ |
| <u>Counting Sort</u> | $\Omega(n+k)$ | $\Theta(n+k)$ | $O(n+k)$ | $O(k)$ |
| <u>Cubesort</u> | $\Omega(n)$ | $\Theta(n \log(n))$ | $O(n \log(n))$ | $O(n)$ |

Learn More

[Cracking the Coding Interview: 150 Programming Questions and Solutions](#)

[Introduction to Algorithms, 3rd Edition](#)

[Data Structures and Algorithms in Java \(2nd Edition\)](#)

[High Performance JavaScript \(Build Faster Web Application Interfaces\)](#)

Get the Official Big-O Cheat Sheet Poster



Josh Davis

Max Hoffmann

Vinnie Magro

Brandon Amos

Apurva K