

57.5k
Shares

Tweet

Share

Share

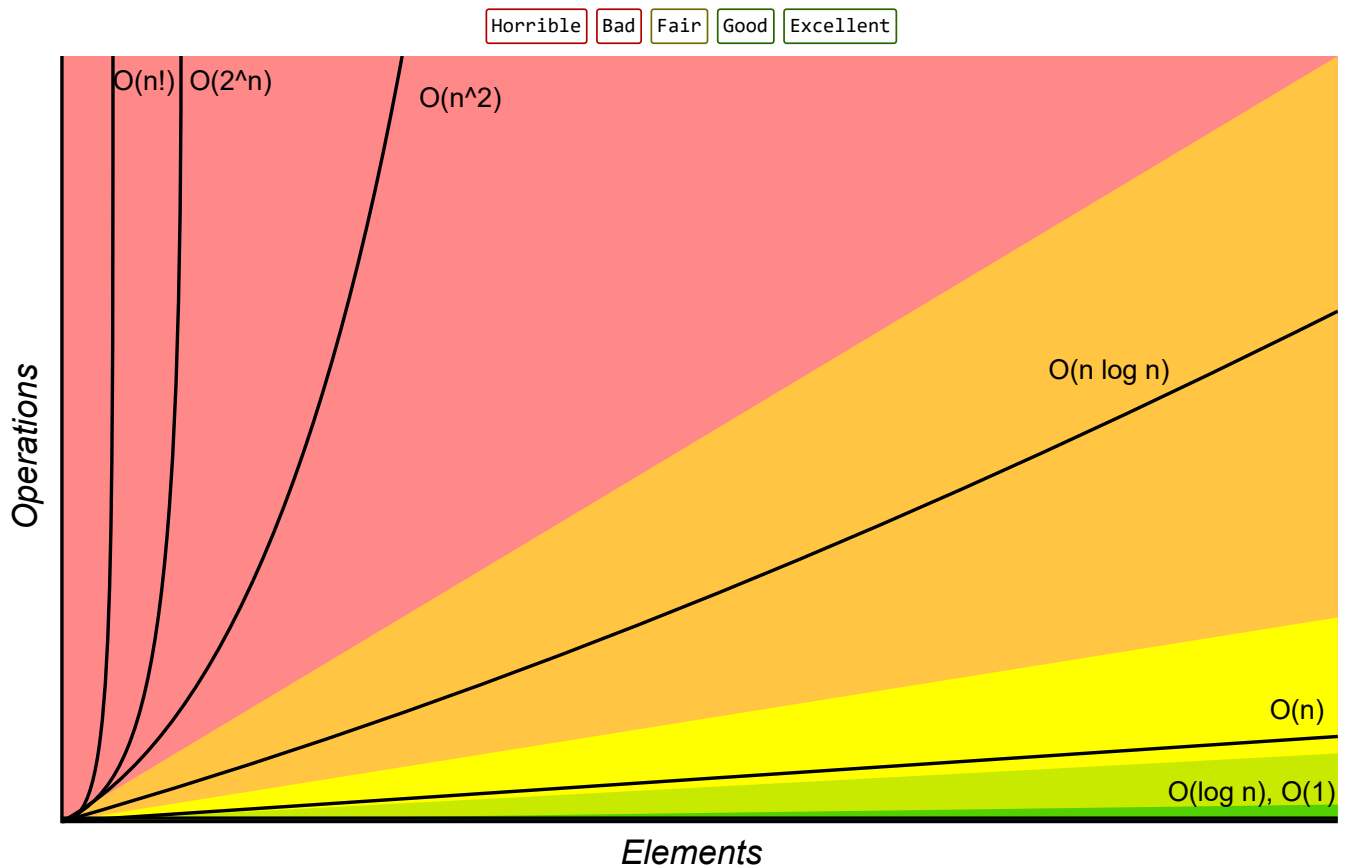
Share

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 Uber, 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](#)

[Check out El Grapho, a graph data visualization library that supports millions of nodes and edges](#)

Big-O Complexity Chart



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

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
<u>Quicksort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n^2)$	$\Theta(\log(n))$
<u>Mergesort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n)$
<u>Timsort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n)$
<u>Heapsort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n \log(n))$	$\Theta(1)$
<u>Bubble Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$
<u>Insertion Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$
<u>Selection Sort</u>	$\Omega(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$
<u>Tree Sort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n^2)$	$\Theta(n)$
<u>Shell Sort</u>	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$\Theta(n(\log(n))^2)$	$\Theta(1)$
<u>Bucket Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$\Theta(n^2)$	$\Theta(n)$
<u>Radix Sort</u>	$\Omega(nk)$	$\Theta(nk)$	$\Theta(nk)$	$\Theta(n+k)$
<u>Counting Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$\Theta(n+k)$	$\Theta(k)$
<u>Cubesort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$\Theta(n \log(n))$	$\Theta(n)$

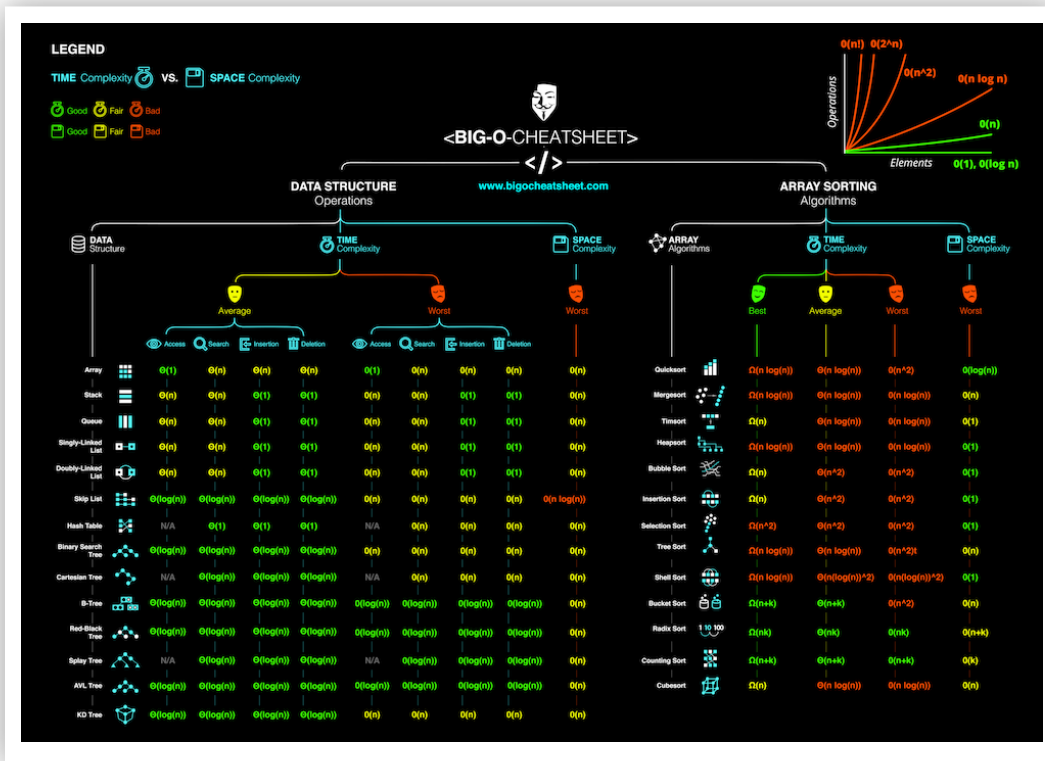
Learn More

[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



Contributors

Eric Rowell	Quentin Pleple	Michael Abed	Nick Dizazzo	Adam Forsyth	Felix Zhu	Jay Engineer
Josh Davis	Nodir Turakulov	Jennifer Hamon	David Dorfman	Bart Massey	Ray Pereda	Si Pham
Mike Davis	mcverry	Max Hoffmann	Bahador Saket	Damon Davison	Alvin Wan	Alan Briolat
Drew Hannay	Andrew Rasmussen	Dennis Tsang	Vinnie Magro	Adam Arold	Alejandro Ramirez	
Aneel Nazareth	Rahul Chowdhury	Jonathan McElroy	steven41292	Brandon Amos	Joel Friedly	
Casper Van Gheluwe	Eric Lefevre-Ardant	Oleg	Renfred Harper	Piper Chester	Miguel Amigot	Apurva K
Matthew Daronco	Yun-Cheng Lin	Clay Tyler	Orhan Can Ozalp	Ayman Singh	David Morton	
Aurelien Ooms	Sebastian Paaske Torholm	Koushik Krishnan	Drew Bailey	Robert Burke		

Make this Page Better

Edit these tables!

499 Comments

Big-O Cheat Sheet



1 Login

Recommend 327

Tweet

Share

Sort by Best