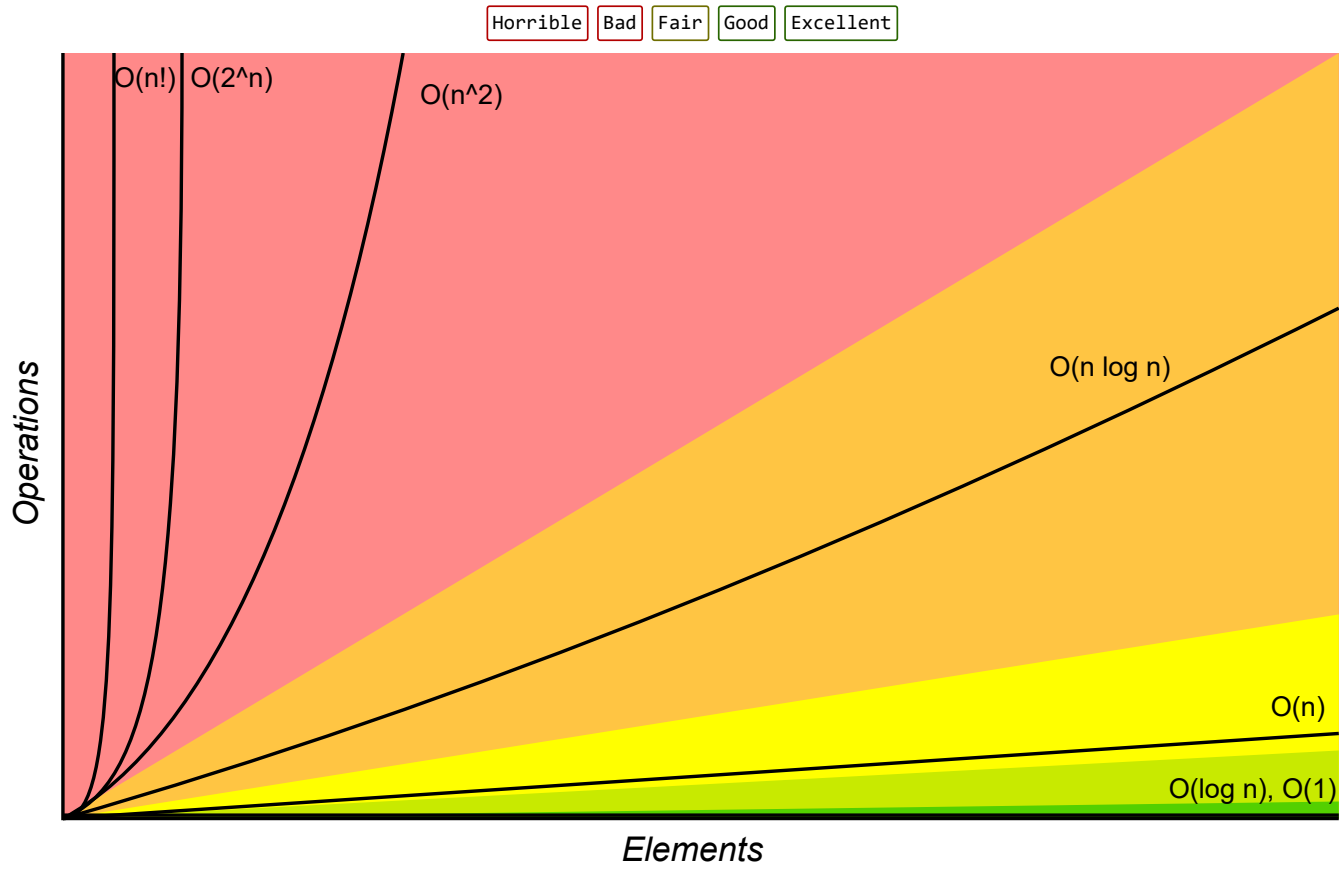
Know Thy Complexities!

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

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	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

Learn More

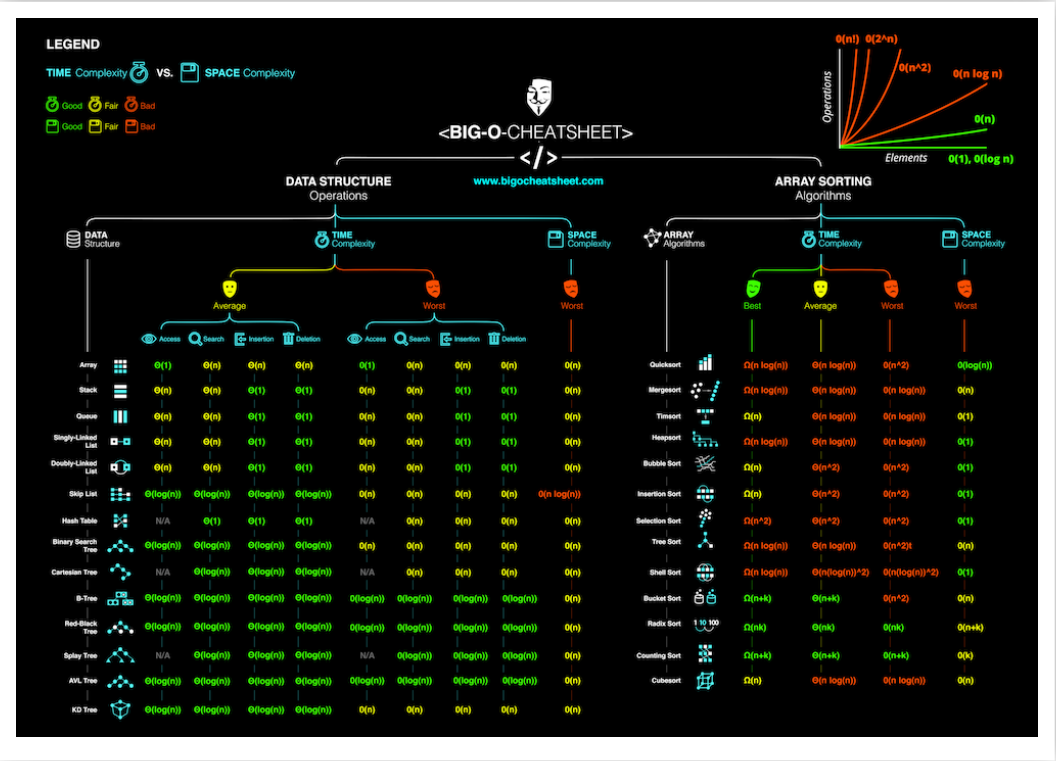
Cracking the Coding Interview: 150 Programming Questions and Solutions

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



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!

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS

Name

Michael Mitchell • 5 years ago

This is great. Maybe you could include some resources (links to khan academy, mooc etc) that would explain each of these concepts for people trying to learn them.

317 • Reply • Share ›

Amanda Harlin ↗ Michael Mitchell • 5 years ago

Yes! Please & thank you

72 • Reply • Share ›

Asim Ahmad ↗ Amanda Harlin • 2 months ago

Can you Explain the Above Algorithm.??

• Reply • Share ›

Cam Cecil ↗ Michael Mitchell • 5 years ago

This explanation in 'plain English' helps: http://stackoverflow.com/qu...

32 • Reply • Share ›

Richard Wheatley ↗ Cam Cecil • 2 years ago

this is plain english.

4 ^ | v • Reply • Share ›

Arjan Nieuwenhuizen ➚ Michael Mitchell • 5 years ago

Here are the links that I know of.

#1) http://aduni.org/courses/al...
#2) http://ocw.mit.edu/courses/...
#3) https://www.udacity.com/cou...

probably as good or maybe better # 2, but I have not had a chance to look at it.
http://ocw.mit.edu/courses/...

Sincerely,
Arjan

p.s.
https://www.coursera.org/co...
This course has just begun on coursera (dated 1 July 2013), and looks very good.

16 ^ | v • Reply • Share ›

fireheron ➚ Arjan Nieuwenhuizen • 4 years ago

Thank you Arjan. Espaecially the coursera.org one ;-)

3 ^ | v • Reply • Share ›

@hangtwentyy ➚ fireheron • 3 years ago

also this! http://opendatastructures.org

6 ^ | v • Reply • Share ›

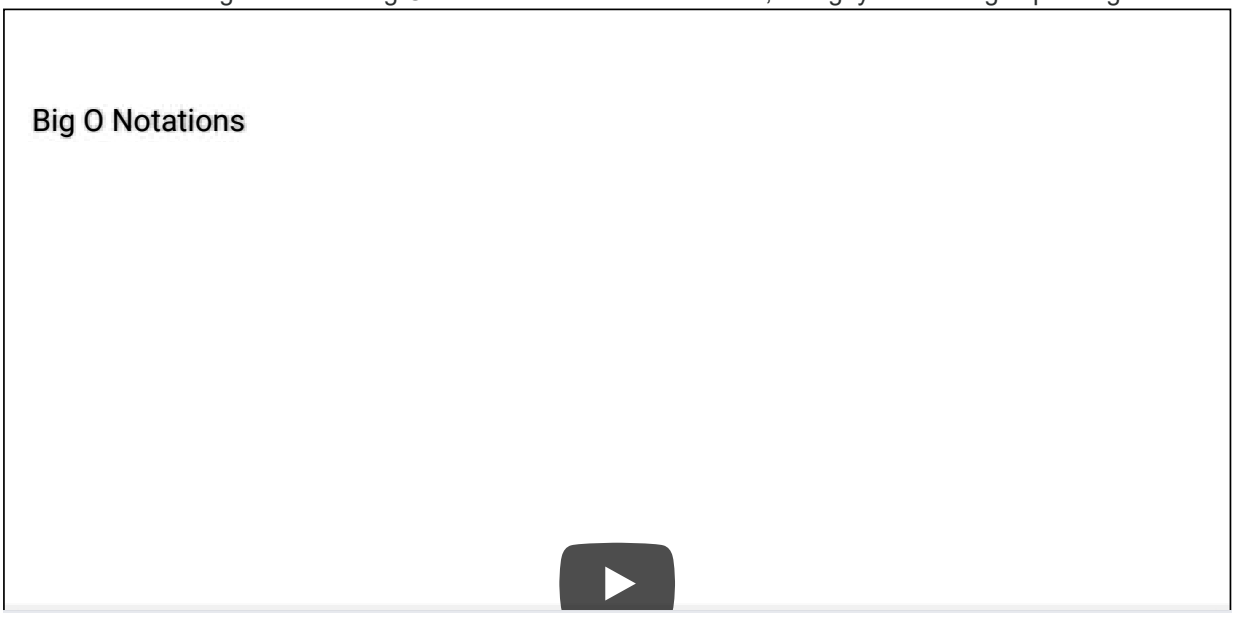
yth ➚ @hangtwentyy • 3 years ago

thank you for sharing this.

1 ^ | v • Reply • Share ›

Eduardo Sánchez ➚ Michael Mitchell • 2 years ago

There is an amazing tutorial for Big O form Derek Banas in Youtube, that guy is amazing explaining!!!



see more

6 ^ | v • Reply • Share ›