



Big o Cheatsheet - Data structures and Algorithms with thiercomplexities

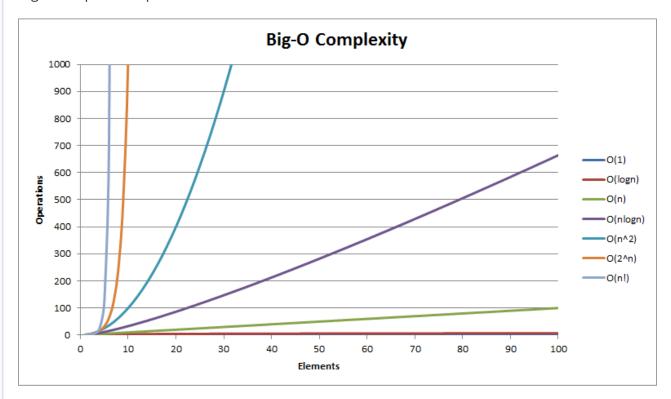
Time-complexity

Algorithms

Big-o

Big o cheatsheet with complexities chart

Big o complete Graph



Legend











Data Structures

Data Structure	Time Comple		Space Complexity						
	Average				Worst				Worst
	Indexing	Search	Insertion	Deletion	Indexing	Search	Insertion	Deletion	
Basic Array	0(1)	0(n)			0(1)	0(n)	-		0(n)
Dynamic Array	0(1)	0(n)	O(n)	0(n)	0(1)	O(n)	O(n)	0(n)	0(n)
Singly-Linked List	O(n)	0(n)	0(1)	0(1)	O(n)	O(n)	0(1)	0(1)	0(n)
Doubly-Linked List	0(n)	0(n)	0(1)	0(1)	0(n)	0(n)	0(1)	0(1)	0(n)
Skip List	O(log(n))	0(log(n))	0(log(n))	O(log(n))	0(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	-	0(1)	0(1)	0(1)		O(n)	O(n)	0(n)	0(n)
Binary Search Tree	O(log(n))	0(log(n))	0(log(n))	0(log(n))	O(n)	0(n)	O(n)	0(n)	0(n)
Cartresian Tree		0(log(n))	O(log(n))	0(log(n))		O(n)	0(n)	0(n)	0(n)
B-Tree	O(log(n))	0(log(n))	O(log(n))	O(log(n))	O(log(n))	0(log(n))	O(log(n))	0(log(n))	0(n)
Red-Black Tree	O(log(n))	0(log(n))	0(log(n))	0(log(n))	O(log(n))	0(log(n))	0(log(n))	0(log(n))	0(n)
Splay Tree		0(log(n))	0(log(n))	0(log(n))		0(log(n))	0(log(n))	0(log(n))	0(n)
AVL Tree	0(log(n))	O(log(n))	O(log(n))	O(log(n))	0(log(n))	O(log(n))	O(log(n))	O(log(n))	0(n)

Data Structures

Data Structure	Time Comple	Space Complexity							
	Average				Worst				Worst
	Indexing	Search	Insertion	Deletion	Indexing	Search	Insertion	Deletion	
Basic Array	0(1)	0(n)			0(1)	0(n)			0(n)
Dynamic Array	0(1)	O(n)	O(n)	O(n)	0(1)	0(n)	O(n)	O(n)	O(n)
Singly-Linked List	0(n)	0(n)	0(1)	0(1)	0(n)	0(n)	0(1)	0(1)	0(n)
Doubly-Linked List	0(n)	0(n)	0(1)	0(1)	0(n)	0(n)	0(1)	0(1)	0(n)
Skip List	0(log(n))	0(log(n))	0(log(n))	O(log(n))	0(n)	0(n)	0(n)	0(n)	O(n log(n))
Hash Table	•	0(1)	0(1)	0(1)		0(n)	O(n)	0(n)	O(n)
Binary Search Tree	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(n)	0(n)	(n)	0(n)	0(n)
Cartresian Tree	-	0(log(n))	O(log(n))	O(log(n))		0(n)	0(n)	0(n)	0(n)
B-Tree	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(n)
Red-Black Tree	O(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(n)
Splay Tree	-	0(log(n))	O(log(n))	0(log(n))		0(log(n))	0(log(n))	0(log(n))	0(n)
AVL Tree	0(log(n))	0(log(n))	O(log(n))	0(log(n))	0(log(n))	O(log(n))	O(log(n))	0(log(n))	0(n)

Searching

Algorithm	Data Structure	Time Complexity		Space Complexity
		Average	Worst	Worst
Depth First Search (DFS)	Graph of V vertices and E edges		O(E + V)	0(V)
Breadth First Search (BFS)	Graph of V vertices and E edges	-	O(E + V)	0(V)
Binary search	Sorted array of n elements	0(log(n))	O(log(n))	0(1)
Linear (Brute Force)	Array	O(n)	0(n)	0(1)
Shortest path by Dijkstra, using a Min-heap as priority queue	Graph with V vertices and E edges	0((V + E) log V)	O((V + E) log V)	0(V)
Shortest path by Dijkstra, using an unsorted array as priority queue	Graph with V vertices and E edges	0(V ^2)	0(V ^2)	(v)
Shortest path by Bellman-Ford	Graph with V vertices and E edges	0(V E)	O(V E)	0(v)

Sorting Algorithms chart

Sorting

Algorithm	Data Structure	Time Complexit	у	Worst Case Auxiliary Space Complexity	
		Best	Average	Worst	Worst
Quicksort	Array	O(n log(n))	O(n log(n))	0(n^2)	O(n)
Mergesort	Array	O(n log(n))	O(n log(n))	O(n log(n))	O(n)
Heapsort	Array	O(n log(n))	O(n log(n))	0(n log(n))	0(1)
Bubble Sort	Array	0(n)	O(n^2)	0(n^2)	0(1)
Insertion Sort	Array	O(n)	O(n^2)	0(n^2)	0(1)
Select Sort	Array	O(n^2)	O(n^2)	O(n^2)	0(1)
Bucket Sort	Array	0(n+k)	O(n+k)	0(n^2)	O(nk)
Radix Sort	Array	O(nk)	O(nk)	0(nk)	0(n+k)

Heaps

Heaps	Time Complexity								
	Heapify	Find Max	Extract Max	Increase Key	Insert	Delete	Merge		
Linked List (sorted)		0(1)	0(1)	O(n)	0(n)	0(1)	O(m+n)		
Linked List (unsorted)	-	0(n)	O(n)	0(1)	0(1)	0(1)	0(1)		
Binary Heap	O(n)	0(1)	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(m+n)		
Binomial Heap		O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))		
Fibonacci Heap	<u> </u>	0(1)	0(log(n))*	0(1)*	0(1)	0(log(n))*	0(1)		

Graphs

Node / Edge Management	Storage	Add Vertex	Add Edge	Remove Vertex	Remove Edge	Query
Adjacency list	O(V + E)	0(1)	0(1)	O(V + E)	O(E)	0(V)
Incidence list	0(V + E)	0(1)	0(1)	O(E)	O(E)	O(E)
Adjacency matrix	0(V ^2)	0(V ^2)	0(1)	0(V ^2)	0(1)	0(1)
Incidence matrix	0(V + E)	O(V + E)	O(V + E)	O(V - E)	O(V + E)	O(E)

Like 8 Tweet

.

AUTHOR



Varun N R

Frontend Developer at Loll...

Pangalore, Karnataka, India

2 notes

TRENDING NOTES

Python Diaries Chapter 3 Map | Filter | Forelse | List Comprehension written by Divyanshu Bansal

Bokeh | Interactive Visualization Library | Use Graph with Django Template written by Prateek Kumar

Bokeh | Interactive Visualization Library | Graph Plotting written by Prateek Kumar

Python Diaries chapter 2 written by Divyanshu Bansal

Python Diaries chapter 1 written by Divyanshu Bansal

more ...

	Resources	Solutions	Company	yService & Support		
	Tech Recruitment	Assess Developers	About Us			
	Blog	Conduct Remote	Press	Technical Support		
+1-650-461-4192	Product Guides	Interviews	Careers	Contact Us		
contact@hackerearth.co	Developer hiring ^m guide	Assess University Talent				
	Engineering Blog	Organize Hackathons				
f y in	Developers Blog					

7

Developers Wiki

Competitive Programming Start a Programming

Club

Practice Machine

Learning

Site Language: English 🕴 | © 2020 HackerEarth All rights reserved | Terms of Service | Privacy Policy