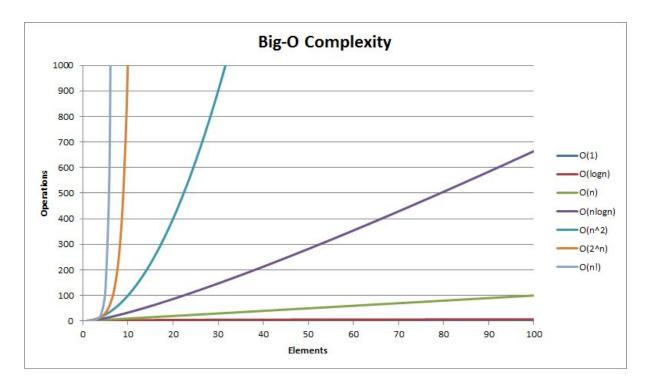
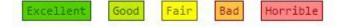
Big O Cheat Sheet

Data structures and Algorithms with their Complexities

• Big O Complete Graph



Legend



Data Structures

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

Data Structures

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

Searching

Algorithm	Data Structure	Time Complexity		Space Complexity
		Average	Worst	Worst
Depth First Search (DFS)	Graph of V vertices and E edges		0(E + V)	(A)
Breadth First Search (BFS)	Graph of V vertices and E edges		0(E + V)	0(v)
Binary search	Sorted array of n elements	O(log(n))	0(log(n))	0(1)
Linear (Brute Force)	Array	(u)0	(u)0	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)	0((V + E) log V)	0(۸)
Shortest path by Dijkstra, using an unsorted array as priority queue	Graph with V vertices and E edges	0(V ^2)	0(v ^2)	0(۸)0
Shortest path by Bellman-Ford	Graph with [V] vertices and [E] edges	0(V E)	0(v E)	(\nu)

Sorting

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

Heaps

Heaps	Time Complexity	exity					
	Heapify	Find Max	Extract Max	Increase Key	Insert	Delete	Merge
Linked List (sorted)		0(1)	0(1)	(u))	(u)0	0(1)	0(m+n)
Linked List (unsorted)		(u)0	(u)0	0(1)	0(1)	0(1)	(1)0
Binary Heap	(u)0	0(1)	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(m+n)
Binomial Heap		0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))	0(log(n))
Fibonacci Heap	(•)	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	0(V + E)	0(1)	0(1)	0(V + E)	0(E)	0(v)
Incidence list	0(V + E)	0(1)	0(1)	0(E)	0(E)	0(E)
Adjacency matrix	0(V ^2)	0(V ^2)	0(1)	0(V ^2)	0(1)	0(1)
Incidence matrix	0(V : E)	0(v · E)	0(v · E)	0(V · E)	0(v · E)	0(E)