Cheat Sheet

This is a compilation of worst-case complexities for various data-structures and algorithms.

Data-Structures

Data Structure	Worst Case Complexity	Notes
Array	Insert: *O(1)* Retrieve: *O(1)*	

Linked List	Insert at Tail	*O(n)*	Note that if new elements are added at the head of the linkedlist then insert becomes a O(1) operation.
	Insert at Head	*O(1)*	
	Retrieve	*O(n)*	
Binary Tree	Insert	*O(n)*	In worst case, the binary tree becomes a linked-list.
	Retrieve	*O(n)*	

Dynamic Array	Insert $O(1)$ Retrieve $O(1)$	Note by retrieving it is implied we are retrieving from a specific index of the array.
Stack	Push $O(1)$ Pop $O(1)$	There are no complexity trick questions asked for stacks or queues. We only mention them here for completeness. The two data-structures are more important from a last-in last-out (stack) and first in first out (queue) perspective.

Queue	Enqueue	*O(1)*	
	Dequeue	*O(1)*	
Priority Queue (binary heap)	Insert	*O(lgn)*	
	Delete	*O(lgn)*	
	Get Max/Min	*O(1)*	

Hashtable	Insert: *O(n)* Retrieve: *O(n)*	Be mindful that a hashtable's average case for insertion and retrieval is O(1)
B-Trees	Insert: *O(logn)* Retrieve: *O(logn)*	
Red-Black Trees	Insert: *O(logn)* Retrieve: *O(logn)*	

<table>
<tr><th>Category</th><th>Worst Case Complexity</th><th>Notes</th></tr>
<tr><td>Sorting</td><td><table>
<tr><td>Bubble Sort</td><td>$O(n^2)$</td></tr>
<tr><td>Insertion Sort</td><td>$O(n^2)$</td></tr>
<tr><td>Selection Sort</td><td>$O(n^2)$</td></tr>
<tr><td>Quick Sort</td><td>$O(n^2)$</td></tr>
<tr><td>Merge Sort</td><td>$O(nlgn)$</td></tr>
</table></td><td>Note, even though worst case quicksort performance is $O(n^2)$ but in practice quicksort is often used for sorting since its average case is O(nlgn).</td></tr>
</table>

Trees	Depth First Search	*O(n)*	***n*** is the total number of nodes in the tree. Most tree-traversal algorithms will end up seeing every node in the tree and their complexity in the worst case is thus O(n).
	Breadth First Search	*O(n)*	
	Pre-order, In-order, Post-order Traversals	*O(n)*	

Did you complete this lesson?

YES!

Complexity Interview Questions

Data Structures

Report an Issue

Ask a Question
(https://discuss.educative.io/tag/cheat-sheet__the-end__big-o-notation-for-coding-interviews-and-beyond)