Data Structure Operations Cheat Sheet									
Data Structure Name	Average Case Time Complexity				Worst Case Time Complexity				Space Complexity
	Accessing n^{th} element	Search	Insertion	Deletion	Accessing n^{th} element	Search	Insertion	Deletion	Worst Case
Arrays	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stacks	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queues	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Binary Trees	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
Binary Search Trees	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(n)	O(n)	O(n)	O(n)	O(n)
Balanced Binary Search Trees	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)	O(*log* n)
Hash Tables	N/A	O(1)	O(1)	O(1)	N/A	O(n)	O(n)	O(n)	O(n)

Note: For best case operations, the time complexities are O(1)

Sorting Algorithms Cheat Sheet							
Sorting Algorithm Name	Time Complexity			Space Complexity	Is Stable?	Sorting Class Type	Remarks
	Best Case	Average Case	Worst Case	Worst Case			
Bubble Sort	O(n)	$O(n^2)$	$O(n^2)$	O(1)	Yes	Comparison	Not a preferred sorting algortihm.
Insetion Sort	O(n)	$O(n^2)$	$O(n^2)$	O(1)	Yes	Comparison	In the best case(already sorted), every insert requires constant time.
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	O(1)	Yes	Comparison	Even a perfectly sorted array requires scanning the entire array.
Merge Sort	O(n*log* n)	O(n*log* n)	O(n*log* n)	O(n)	Yes	Comparison	On arrays, it requiers O(n) space; and on linked list, it requires constant space.
Heap Sort	O(n*log* n)	O(n*log* n)	O(n*log* n)	O(1)	No	Comparison	By using input array as storage for the heap, it is possible to achieve constant space.
Quick Sort	O(n*log* n)	O(n*log* n)	$O(n^2)$	O(n*log* n)	No	Comparison	Randomly picking a pivot value can help avoid worst case scenarios such as a perfectly sorted array.
Tree Sort	O(n*log* n)	O(n*log* n)	$O(n^2)$	O(n)	Yes	Comparison	Performing inorder traversal on the balanced binary search tree.
Counting Sort	O(n + k)	O(n + k)	O(n + k)	O(k)	Yes	Linear	Where *k* is the range of the non-negative key values.
Bucket Sort	O(n + k)	O(n + k)	$O(n^2)$	O(n)	Yes	Linear	Bucket sort is stable, if the underlying sorting algorithm is stable.
Radix Sort	O(dn)	O(dn)	O(dn)	O(d + n)	Yes	Linear	Radix sort is stable, if the underlying sorting algorithm is stable.