Cheat Sheet

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

Data-Structures

<table>
<tr><th>Data Structure</th><th>Worst Case Complexity</th><th>Notes</th></tr>
<tr><td>Array</td><td><table><tr><td>Insert</td><td>O(1)</td></tr><tr><td>Retrieve</td><td>O(1)</td></tr></table></td><td rowspan="2">Note that if new elements are added at the head of the linkedlist then insert becomes a O(1) operation.</td></tr>
<tr><td>Linked List</td><td><table><tr><td>Insert at Tail</td><td>O(n)</td></tr><tr><td>Insert at Head</td><td>O(1)</td></tr><tr><td>Retrieve</td><td>O(n)</td></tr></table></td></tr>
<tr><td>Binary Tree</td><td><table><tr><td>Insert</td><td>O(n)</td></tr><tr><td>Retrieve</td><td>O(n)</td></tr></table></td><td>In worst case, the</td></tr>
</table>

	Insert	*O(n)*	binary tree becomes a linked-list.
	Retrieve	*O(n)*	
Dynamic Array	Insert	*O(1)*	Note by retrieving it is implied we are retrieving from a specific index of the array.
	Retrieve	*O(1)*	
Stack	Push	*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.
	Pop	*O(1)*	
Queue	Enqueue	*O(1)*	
	Dequeue	*O(1)*	

Priority Queue (binary heap)	Insert Delete Get Max/Min	*O(lgn)* *O(lgn)* *O(1)*	
Hashtable	Insert Retrieve	*O(n)* *O(n)*	Be mindful that a hashtable's average case for insertion and retrieval is O(1)
B-Trees	Insert Retrieve	*O(logn)* *O(logn)*	
Red-Black Trees	Insert Retrieve	*O(logn)* *O(logn)*	

Algorithms

<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(n²)</td></tr>
<tr><td>Insertion Sort</td><td>O(n²)</td></tr>
<tr><td>Selection Sort</td><td>O(n²)</td></tr>
<tr><td>Quick Sort</td><td>O(n²)</td></tr>
<tr><td>Merge Sort</td><td>O(nlgn)</td></tr>
</table></td><td>Note, even though worst case quicksort performance is O(n²) but in practice quicksort is often used for sorting since its average case is O(nlgn).</td></tr>
<tr><td>Trees</td><td><table>
<tr><td>Depth First Search</td><td>O(n)</td></tr>
<tr><td>Breadth First Search</td><td>O(n)</td></tr>
<tr><td>Pre-order,</td><td>O(n)</td></tr>
</table></td><td>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).</td></tr>
</table>

		In- order, Post- order Traver sals	