Runtimes Cheat Sheet

Big-Oh Notation Rules

1. Ignore constant coefficients.
2. Ignore lower-order terms.
3. Can add and multiply.

Formally,

$$f(x) \in O(g(x)) \text{ as } x \to \infty \iff \exists C \in \mathbb{R}^+ \text{ and } \exists x_0 \in \mathbb{R} \text{ such that } \forall x \in \mathbb{R} \text{ where } x \geq x_0, |f(x)| \leq C * g(x).$$

Common Runtimes

Runtimes are listed in order of increasing complexity. Note that this is not an exhaustive list.

$O(1), O(\log n), O(n), O(n \log n), O(n^2), O(n^3), ..., O(n^k), ..., O(2^n), O(3^n), ..., O(k^n), ..., O(n!), ...$

Runtimes of Data Structures and Algorithms

Stacks

`push(item)`	$O(1)$
`pop()`	$O(1)$
`peek()`	$O(1)$

Queues

`Enqueue(item)`	$O(1)$
`Dequeue()`	$O(1)$
`Peek()`	$O(1)$

Binary Search Trees

	Balanced	Unbalanced
`Insert(key, value)`	$O(\log n)$	$O(n)$
`Remove(key)`	$O(\log n)$	$O(n)$
`Lookup(key)`	$O(\log n)$	$O(n)$

Heaps and Priority Queues

`insert(key, value)`	$O(\log n)$
`extract()`	$O(\log n)$
`peek()`	$O(1)$
`bubbleup(index)`	$O(\log n)$
`bubbledown(index)`	$O(\log n)$
`heapify(list)`	$O(n)$
`merge(list)`	$O(n)$
`get_priority(item)`	$O(1)$
`update_priority(item, priority)`	$O(\log n)$
`remove(item)`	$O(\log n)$

Trees

Note that order can be one of Preorder, Inorder, or Postorder.

`traverse(order)`	$O(n)$

Graphs

Note that `weighted_shortest_paths()` is Dijkstra's Algorithm.

		Requirements
`breadth_first_search(start_node)`	$O(V + E)$	Any
`depth_first_search()`	$O(V + E)$	Any
`topological_sort()`	$O(V + E)$	Directed Acyclic Graph
`weighted_shortest_paths()`	$O(E + V \log V)$	Weighted Graph

Sorting

	Best Case	Average Case	Worst Case
`bubble_sort(list)`	$O(n)$	$O(n^2)$	$O(n^2)$
`selection_sort(list)`	$O(n^2)$	$O(n^2)$	$O(n^2)$
`insertion_sort(list)`	$O(n)$	$O(n^2)$	$O(n^2)$
`merge_sort(list)`	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
`quicksort(list)`	$O(n \log n)$	$O(n \log n)$	$O(n^2)$
`heapsort(list)`	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
`introsort(list)`	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$