

Data Structures Cheat Sheet

For coding interviews or computer science classes

A quick reference of the big O costs and core properties of every data structure.

Time Complexity of Common Operations

	index	search	insert	delete
Array	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Dynamic Array	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Linked List	$O(n)$	$O(n)$	$O(n)$	$O(1)$
Queue	$O(1)$	$O(n)$	$O(1)$	$O(1)$
Stack	$O(1)$	$O(n)$	$O(1)$	$O(1)$
Hash Table	—	$O(1)$	$O(1)$	$O(1)$
Binary Search Tree	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$
Trie	—	$O(m)$	$O(m)$	$O(m)$
Heap	$O(n)$	$O(n)$	$O(\lg n)$	$O(n)$
Priority Queue	$O(n)$	$O(n)$	$O(\lg n)$	$O(1)$
Bloom filter	—	$O(1)$	$O(1)$	—
LRU Cache	—	$O(1)$	$O(1)$	$O(1)$

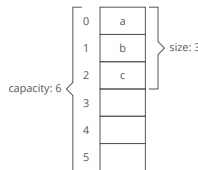
Array

Stores things in order. Has quick lookups by index.



Dynamic Array

An array that automatically grows as you add more items.



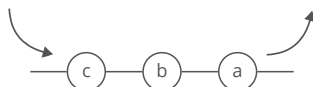
Linked List

Also stores things in order. Faster insertions and deletions than arrays, but slower lookups (you have to "walk down" the whole list).



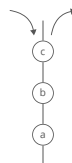
Queue

Like the line outside a busy restaurant. "First come, first served."



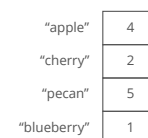
Stack

Like a stack of dirty plates in the sink. The first one you take off the top is the last one you put down.



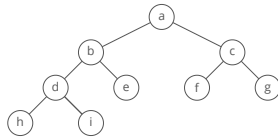
Hash Table

Like an array, except instead of indices you can set arbitrary keys for each value.



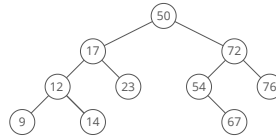
Tree

Good for storing hierarchies. Each node can have "child" nodes.



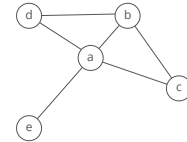
Binary Search Tree

Everything in the left subtree is smaller than the current node, everything in the right subtree is larger. $O(\lg n)$ lookups, but only if the tree is balanced!



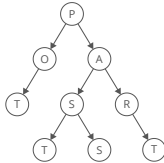
Graph

Good for storing networks, geography, social relationships, etc.



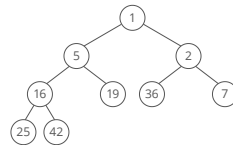
Trie

Stores a set of strings in a big tree of characters. Good for lookups by prefix. Sometimes saves space.



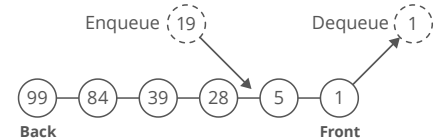
Heap

A binary tree where the smallest value is always at the top. Use it to implement a priority queue.



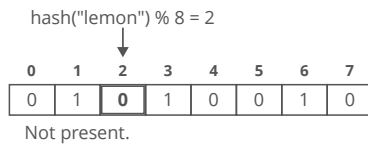
Priority Queue

A queue where items are ordered by priority.



Bloom filter

A constant-space bitmap that lets you quickly check whether or not an item is in a set. Can give false positives.



LRU Cache

Lets you quickly identify which item hasn't been used for the longest amount of time.

