**Know Thy Complexities!**

Hi there!  This webpage covers the space and time Big-O complexities of common algorithms used in Computer Science.  When preparing for technical interviews in the past, I found myself spending hours crawling the internet putting together the best, average, and worst case complexities for search and sorting algorithms so that I wouldn't be stumped when asked about them.  Over the last few years, I've interviewed at several Silicon Valley startups, and also some bigger companies, like Yahoo, eBay, LinkedIn, and Google, and each time that I prepared for an interview, I thought to myself "Why hasn't someone created a nice Big-O cheat sheet?".  So, to save all of you fine folks a ton of time, I went ahead and created one.  Enjoy! - [Eric](https://twitter.com/ericdrowell)

**Legend**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Excellent | Good | Fair | Bad | Horrible |

**Data Structure Operations**

| **Data Structure** | **Time Complexity** | | | | | | | | **Space Complexity** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Average** | | | | **Worst** | | | | **Worst** |
|  | **Access** | **Search** | **Insertion** | **Deletion** | **Access** | **Search** | **Insertion** | **Deletion** |  |
| [Array](http://en.wikipedia.org/wiki/Array_data_structure) | O(1) | O(n) | O(n) | O(n) | O(1) | O(n) | O(n) | O(n) | O(n) |
| [Stack](http://en.wikipedia.org/wiki/Stack_%28abstract_data_type%29) | O(n) | O(n) | O(1) | O(1) | O(n) | O(n) | O(1) | O(1) | O(n) |
| [Singly-Linked List](http://en.wikipedia.org/wiki/Singly_linked_list#Singly_linked_lists) | O(n) | O(n) | O(1) | O(1) | O(n) | O(n) | O(1) | O(1) | O(n) |
| [Doubly-Linked List](http://en.wikipedia.org/wiki/Doubly_linked_list) | O(n) | O(n) | O(1) | O(1) | O(n) | O(n) | O(1) | O(1) | O(n) |
| [Skip List](http://en.wikipedia.org/wiki/Skip_list) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(n) | O(n) | O(n) | O(n) | O(n log(n)) |
| [Hash Table](http://en.wikipedia.org/wiki/Hash_table) | - | O(1) | O(1) | O(1) | - | O(n) | O(n) | O(n) | O(n) |
| [Binary Search Tree](http://en.wikipedia.org/wiki/Binary_search_tree) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(n) | O(n) | O(n) | O(n) | O(n) |
| [Cartesian Tree](https://en.wikipedia.org/wiki/Cartesian_tree) | - | O(log(n)) | O(log(n)) | O(log(n)) | - | O(n) | O(n) | O(n) | O(n) |
| [B-Tree](http://en.wikipedia.org/wiki/B_tree) | 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(n) |
| [Red-Black Tree](http://en.wikipedia.org/wiki/Red-black_tree) | 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(n) |
| [Splay Tree](https://en.wikipedia.org/wiki/Splay_tree) | - | O(log(n)) | O(log(n)) | O(log(n)) | - | O(log(n)) | O(log(n)) | O(log(n)) | O(n) |
| [AVL Tree](http://en.wikipedia.org/wiki/AVL_tree) | 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(n) |

**Array Sorting Algorithms**

| **Algorithm** | **Time Complexity** | | | **Space Complexity** |
| --- | --- | --- | --- | --- |
|  | **Best** | **Average** | **Worst** | **Worst** |
| [Quicksort](http://en.wikipedia.org/wiki/Quicksort) | O(n log(n)) | O(n log(n)) | O(n^2) | O(log(n)) |
| [Mergesort](http://en.wikipedia.org/wiki/Merge_sort) | O(n log(n)) | O(n log(n)) | O(n log(n)) | O(n) |
| [Timsort](http://en.wikipedia.org/wiki/Timsort) | O(n) | O(n log(n)) | O(n log(n)) | O(n) |
| [Heapsort](http://en.wikipedia.org/wiki/Heapsort) | O(n log(n)) | O(n log(n)) | O(n log(n)) | O(1) |
| [Bubble Sort](http://en.wikipedia.org/wiki/Bubble_sort) | O(n) | O(n^2) | O(n^2) | O(1) |
| [Insertion Sort](http://en.wikipedia.org/wiki/Insertion_sort) | O(n) | O(n^2) | O(n^2) | O(1) |
| [Selection Sort](http://en.wikipedia.org/wiki/Selection_sort) | O(n^2) | O(n^2) | O(n^2) | O(1) |
| [Shell Sort](http://en.wikipedia.org/wiki/Shellsort) | O(n) | O((nlog(n))^2) | O((nlog(n))^2) | O(1) |
| [Bucket Sort](http://en.wikipedia.org/wiki/Bucket_sort) | O(n+k) | O(n+k) | O(n^2) | O(n) |
| [Radix Sort](http://en.wikipedia.org/wiki/Radix_sort) | O(nk) | O(nk) | O(nk) | O(n+k) |

**Graph Operations**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Node / Edge Management** | **Storage** | **Add Vertex** | **Add Edge** | **Remove Vertex** | **Remove Edge** | **Query** |
| [Adjacency list](http://en.wikipedia.org/wiki/Adjacency_list) | O(|V|+|E|) | O(1) | O(1) | O(|V| + |E|) | O(|E|) | O(|V|) |
| [Incidence list](http://en.wikipedia.org/wiki/Incidence_list) | O(|V|+|E|) | O(1) | O(1) | O(|E|) | O(|E|) | O(|E|) |
| [Adjacency matrix](http://en.wikipedia.org/wiki/Adjacency_matrix) | O(|V|^2) | O(|V|^2) | O(1) | O(|V|^2) | O(1) | O(1) |
| [Incidence matrix](http://en.wikipedia.org/wiki/Incidence_matrix) | O(|V| ⋅ |E|) | O(|V| ⋅ |E|) | O(|V| ⋅ |E|) | O(|V| ⋅ |E|) | O(|V| ⋅ |E|) | O(|E|) |

**Heap Operations**

| **Type** | **Time Complexity** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Heapify** | **Find Max** | **Extract Max** | **Increase Key** | **Insert** | **Delete** | **Merge** |  |
| [Linked List (sorted)](http://en.wikipedia.org/wiki/Linked_list) | - | O(1) | O(1) | O(n) | O(n) | O(1) | O(m+n) |  |
| [Linked List (unsorted)](http://en.wikipedia.org/wiki/Linked_list) | - | O(n) | O(n) | O(1) | O(1) | O(1) | O(1) |  |
| [Binary Heap](http://en.wikipedia.org/wiki/Binary_heap) | O(n) | O(1) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(m+n) |  |
| [Binomial Heap](http://en.wikipedia.org/wiki/Binomial_heap) | - | O(1) | O(log(n)) | O(log(n)) | O(1) | O(log(n)) | O(log(n)) |  |
| [Fibonacci Heap](http://en.wikipedia.org/wiki/Fibonacci_heap) | - | O(1) | O(log(n)) | O(1) | O(1) | O(log(n)) | O(1) |  |