The Common Complexities Handout

http://HiredInTech.com

HIRED IN TECH

Algorithms

Algorithm	Time Complexity
Binary Search in a sorted array of N elements	$O(\log N)$
Reversing a string of N elements	$O(N)$
Linear search in an unsorted array of N elements	$O(N)$
Compare two strings with lengths L1 and L2	$O(min(L1, L2))$
Computing the Nth Fibonacci number using dynamic programming	$O(N)$
Checking if a string of N characters is a palindrome	$O(N)$
Finding a string in another string using the Aho-Corasick algorithm	$O(N)$
Sorting an array of N elements using Merge Sort/Quick Sort/Heap Sort	$O(N * \log N)$
Sorting an array of N elements using Bubble sort	$O(N^2)$
Two nested loops from 1 to N	$O(N^2)$
The Knapsack problem of N elements with capacity M	$O(N * M)$
Finding a string in another string – the naive approach	$O(L1 * L2)$
Three nested loops from 1 to N	$O(N^3)$
Twenty-eight nested loops … you get the idea	$O(N^{28})$
Generating all subsets of a set of N elements	$O(2^N)$

Data Structures ❖ Stack

Operation	Time Complexity
Adding a value to the top of a stack	$O(1)$
Removing the value at the top of a stack	$O(1)$
Reversing a stack	$O(N)$

Data Structures ❖ Queue

Operation	Time Complexity
Adding a value to end of the queue	$O(1)$
Removing the value at the front of the queue	$O(1)$
Reversing a queue	$O(N)$

Data Structures ❖ Heap

Operation	Time Complexity
Adding a value to the heap	$O(\log N)$
Removing the value at the top of the heap	$O(\log N)$

Data Structures ❖ Hash

Operation	Time Complexity
Adding a value to a hash	$O(1)$
Checking if a value is in a hash	$O(1)$