The Common Complexities Handout

Algorithms

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

Data Structures ❖ Stack

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

Data Structures ❖ Queue

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

Data Structures ❖ Heap

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

Data Structures ❖ Hash

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