

---

# Problem Solving Sessions

---

Mohamed Emary

December 29, 2024

## 1 Sorting Algorithms Runtime

Sorting Algorithms	Time Complexity (Best Case)	Time Complexity (Average Case)	Time Complexity (Worst Case)	Space Complexity (Worst Case)
Bubble Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$\Omega(n^2)$	$\theta(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	$O(1)$
Quick Sort	$\Omega(n \log n)$	$\theta(n \log n)$	$O(n^2)$	$O(n)$
Merge Sort	$\Omega(n \log n)$	$\theta(n \log n)$	$O(n \log n)$	$O(n)$
Heap Sort	$\Omega(n \log n)$	$\theta(n \log n)$	$O(n \log n)$	$O(1)$

## 2 Contains Duplicate

Solution	Time Complexity	Space Complexity
Sorted Array	$O(n \log n)$ Sorting Time Complexity	$O(1)$
Using Hashing (Map in JS)	$O(n)$ Looping on the array	$O(n)$ Space of the hash
Brute Force	$O(n^2)$ Finding whether each item exists or not	$O(n)$ The extra array

## 3 Valid Anagram

## 5 Group Anagram

---

Solution	Time Complexity	Space Complexity
Sorted Array	$O(n \log n)$ Sorting Time Complexity	$O(1)$
Hash Table	$O(n)$	$O(n)$

## 4 Two Sum

Solution	Time Complexity	Space Complexity
Two Pointers	$O(n \log n)$ Sorting for the two pointers technique to work	$O(n)$ for the extra array
Hash Table	$O(m.n \log n)$ String array of length $m$ with strings of length $n$ being sorted in $O(n \log n)$	$O(n)$ for the hash table
Hash Table (Optimized)	$O(n)$ looping on array items	$O(n)$ for the hash

## 5 Group Anagram

Solution	Time Complexity	Space Complexity
Hash Table	$m.n \log n$ String array of length $m$ with strings of length $n$ being sorted in $O(n \log n)$	$O(n)$ for the hash table
Using ASCII Values and Array for counting Appearances of Characters	$O(n.m)$ where $n$ is the length of the array and $m$ is the length of the string	$O(n)$ for the array