

Time Complexities.

- Amount of time taken by an algo to run as a fⁿ of length of the input.
- It is the time taken by CPU to perform a program.

Ex: loop for array of size n:

$$T.C \rightarrow O(n)$$

Space complexity.

- Amount of space taken by an algo to perform a given input of some length.
- It is the space taken by the code to run multiple outputs.

Ex:

int a = 1

→ variable defined.
↓
S.C → $O(1)$.

Units to represent complexities.

1. Big O; upper bound
2. Theta O. Avg case

2. Θ notation

3. Omega Ω ; Lower bound

↓
The lowest time/space needed in a code.

→ worst case of T.C & S.C. But will study this only.

Different types of O (Big O) complexities.

- Constant time: $O(1)$ → $n=5$ int $a=5$
- Linear time: $O(n)$ → 1 loop.
- Logarithmic time: $O(\log n)$ → merge sort [$n \log n$]
- Quadratic time: $O(n^2)$ → 2 loop [Nested]
- Cubic time: $O(n^3)$ → 3 loops [nested].

Hierarchy of Complexity;

$$O(1) < O(\log n) < O(\sqrt{n}) < O(n) < O(n \log n) \\ < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)$$

Q that will be asked by interviewer:

→ If you have a program using $O(n)$ complexity. what are the better options left?

Dutch-National flag problem [Leet: 75]
Sort Colors.

75. Sort Colors Solved

Medium Topics Companies Hint

Given an array `nums` with `n` objects colored red, white, or blue, sort them **in-place** so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers `0`, `1`, and `2` to represent the color red, white, and blue respectively.

You must solve this problem without using the library's sort function.

Example 1:
Input: `nums = [2,0,2,1,1,0]`
Output: `[0,0,1,1,2,2]`

Example 2:
Input: `nums = [2,0,1]`
Output: `[0,1,2]`

Constraints:

- `n == nums.length`
- `1 <= n <= 300`

Handwritten notes:

- Problem Statement
- Don't use library's sort
- Include all Lib.
- The code will be done here
- you don't need to return.
- cm.

```
class Solution {
public:
    void sortColors(vector<int>& nums) {
    }
};
```

20.2K 521 249 Online Testcase Test Result

Problem statement.

There is a given array with n objects colored red - white - blue. Sort them in place so that object of same color adjacent, The order of color should be R - W - B.

R \rightarrow 0

W \rightarrow 1

B \rightarrow 2.

Approach

we have to assign 3 pointers.
 i , left & right.

use while loop ($i \leq \text{right}$).

```
if (nums[i] == 0) {  
    swap(i, left);  
    i++;  
    left;
```

```
}
```

```
else if (nums[i] == 1) {  
    i++;
```

```
}
```

```
else { swap(i, right);  
    i++; right--;
```

