Problem Solving

04 July 2022 20:09

Q. [1, -1, 2, 0, 3, 1, -2]

1+2-1 → 2

3

Triplet

1) 1, -1, 0
2) 2, -2, 0
3) -1, 3, -2
4) 1, 1, -2

1 + (-1) + 0 } => 0
2 + (-2) to
-1 + 3 + (-2)
1 + 1 + (-2)

3 + 2 + ? → 0 (-5)
-(5)

{ i → n-2, j → n-1, k → n }

Set

[1, -1, 2, 0, 3, 1, -2]

i → n-1

j → n

```
for(i=0; i< n-1; i++)
{let Setarray = new Set()
  for(j= i+1; j< n; j++) {
    var Checkno = (arr[i] + arr[j])
    if (Setarray.has(checkno)
    {
        Print. i, j, Checkno
    }
    else
    {
        Setarray.add(arr[j])
```

0 + (-1) → (-1)

l ↓ , r ↓

1	-1	2	0	3	2	-2

x

O(1) O(n^2)

x + arr[l] + arr[r] = 0

1 + (-1) + (-2) = 0
-3 + 1
-2

< 0

> 0

< 0

= 0 } => x, l, r

< 0 } l++

> 0 } r--

3 => O(n^3) => O(n^2).O(n)

2 => O(n^2)

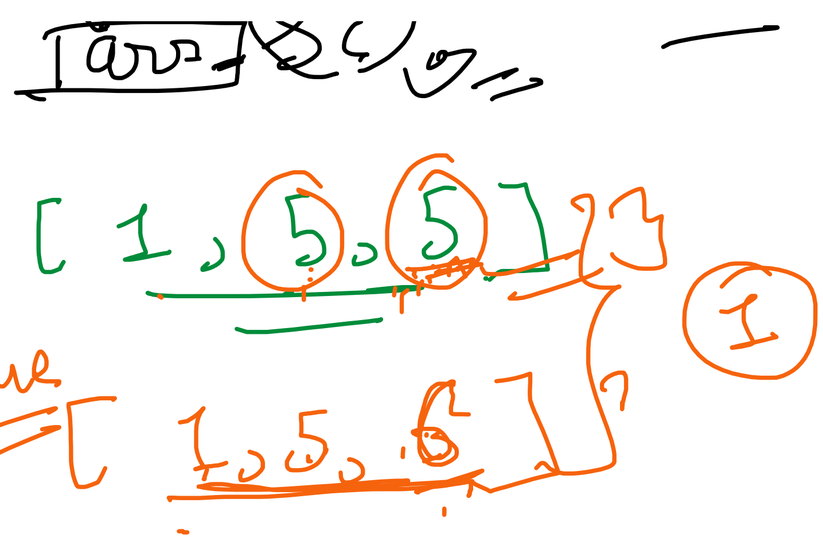
1 => O(n^2) / O(1)

arr.sort() // O(n log n)

O(n^2)

O(n^2)

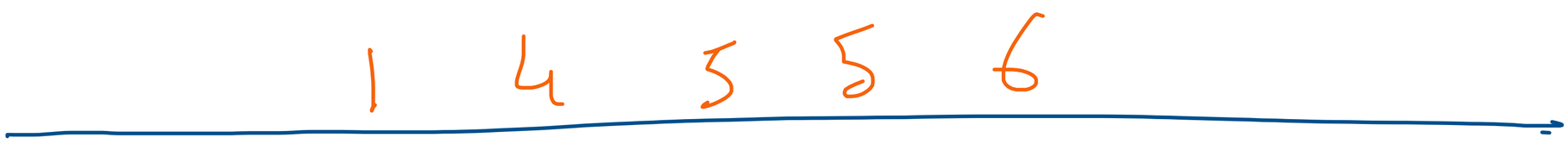
2 Pointer



Problem Statement - Given an array. You need to convert it into a unique array. A unique array is the one where all the element's frequency is 1 i.e. all elements are present only 1 time and there are no duplicates.
[We can increment the number if it is duplicate. This is the move which we can apply to the array]
Our task is to find the number of moves to convert a given array to a unique array.

Input :
Arr = [1,5,5]
Output : 1

As we can increment 5 to 6 in 1 move and our array will be [1, 5 , 6]



Problem 5 : Merge Overlapping Intervals

From a list of time intervals in any order , merge all overlapping intervals , and then return having only mutually exclusive intervals.

Example :

Input : [[1,4] , [2,6] ,[8,10]]
Output : [[1,6] [8,10]]

We can merge [1,4] and [2,6] to [1,6] as there is overlap of [2,4] .