Problem Solving -2

Wednesday, 6 July 2022 6:45 PM

{ Set :. Map }

arr = [1, 2, 3, 4, 5]

const mySet = new Set(arr)

{ [1, 2, 3, 4, 5]
{1, 2, 3, 4, 5}

Set
↳ unique values
Primitive/
Object

arr = [1, 2, 3, 4, 5, 5, 5, 5, 1, 2]

mySet = new Set(arr)

{ 1, 2, 3, 4, 5}

smallest → 0
highest → n-1

arr = [1, 2, 5, 10, 3, 4]

O/P = [0, 1, 4, 5, 2, 3]

0 1 - 0
1 2 - 1
3 5 - 4
3 10 - 5
4 3 - 2
5 4 - 3

orig { 1, 2, 5, 10, 3, 4

map }

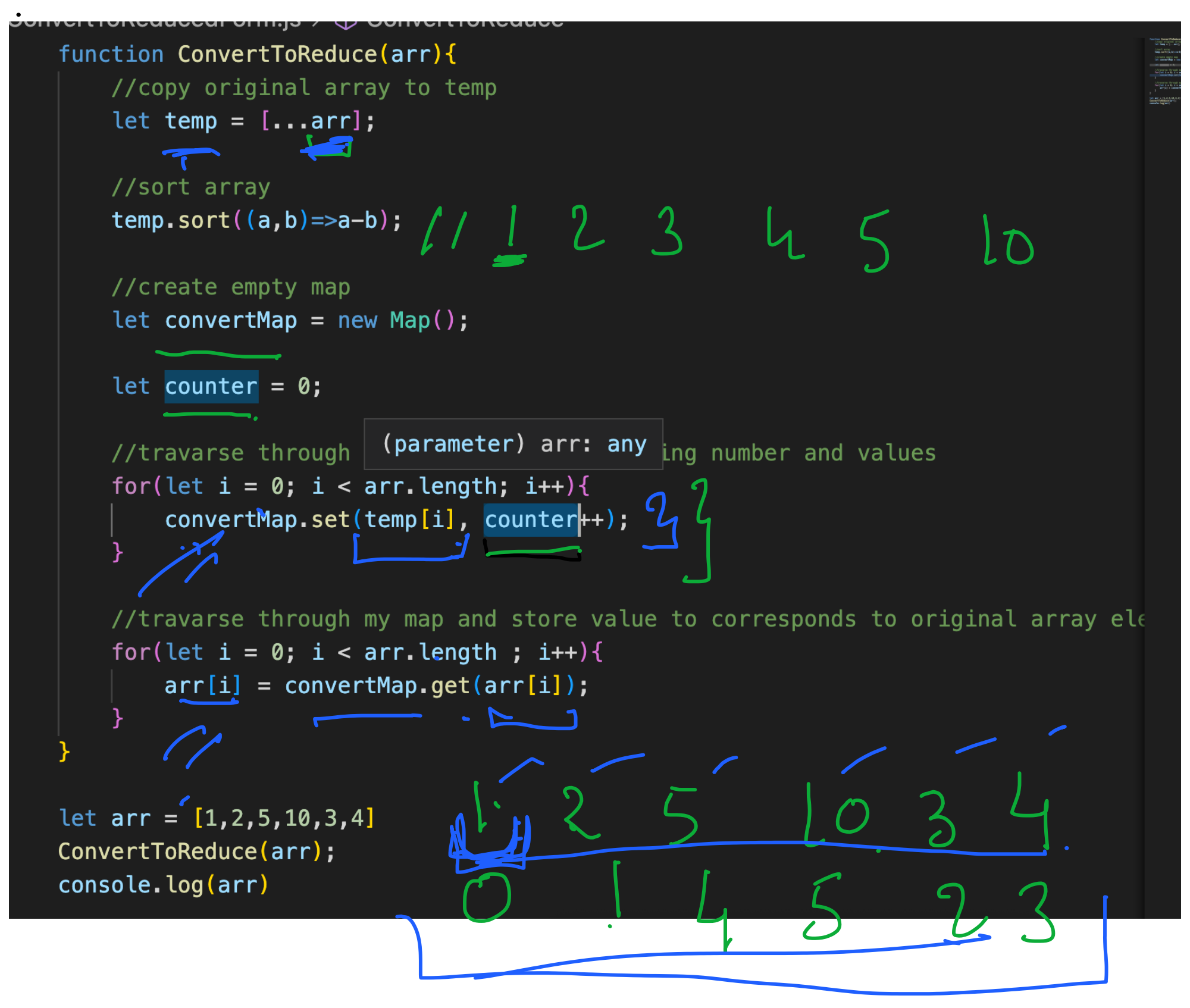
temp array { 1 2 3 4 5 10

count = 0

1. - 0
2 - 1
5 - 4
10 - 3
3 - 2
4 - 3

key - value
1 - 0
2 - 1
3 - 2
4 - 3
5 - 4
10 - 5

.set
.get



```
function ConvertToReduce(arr){
    //copy original array to temp
    let temp = [...arr];

    //sort array
    temp.sort((a,b)=>a-b);

    //create empty map
    let convertMap = new Map();

    let counter = 0;

    //travarse through (parameter) arr: any ing number and values
    for(let i = 0; i < arr.length; i++){
        convertMap.set(temp[i], counter++);
    }

    //travarse through my map and store value to corresponds to original array ele
    for(let i = 0; i < arr.length ; i++){
        arr[i] = convertMap.get(arr[i]);
    }
}

let arr = [1,2,5,10,3,4]
ConvertToReduce(arr);
console.log(arr)
```

// 1 2 3 4 5 10

3 }

1 2 5 10 3 4
0 1 4 5 2 3

Map

key - value
1 - 0
2 - 1
3 - 2
4 - 3
5 - 4
10 - 5

.get(1) // 0

1 7 3 2 1 2 2

Q [1, 1, 2, 2, 3, 7] } logic

↳ nlogn

Sort

1-1+1

0+1

i-1 i

1 → 1 } 2 1<=1

{ 1 2 2 2 3 7

i-1 i

2-2+1

2 <= 2 } true

{ 1 1 2 2 3 7

i-1 i

{ 1 2 3 2 3 7

i-1 i

} increment

2 < = 3

arr[i]-arr(i-1)+1

{ 1 2 3 4 3 7

i-1 i

3 - 2+1
1+1

{ 1 2 3 4 5 7

3 <= 4 true

arr[i] <= arr[i-1]

then

~~counter = counter+~~

arr[i-1] - (

4 -> 3 +1 } 1+1=2

$2+2+1+1 \} = \boxed{6}$

$\underbrace{3 - 2 + 1}_{1+1}$

count[ ] = count[ ] + 1

1 1 2 2 3 7

i-1 i

1 2 2 2 3 7

i-1 i

1 2 3 2 3 7

i-k i

2

counter = counter