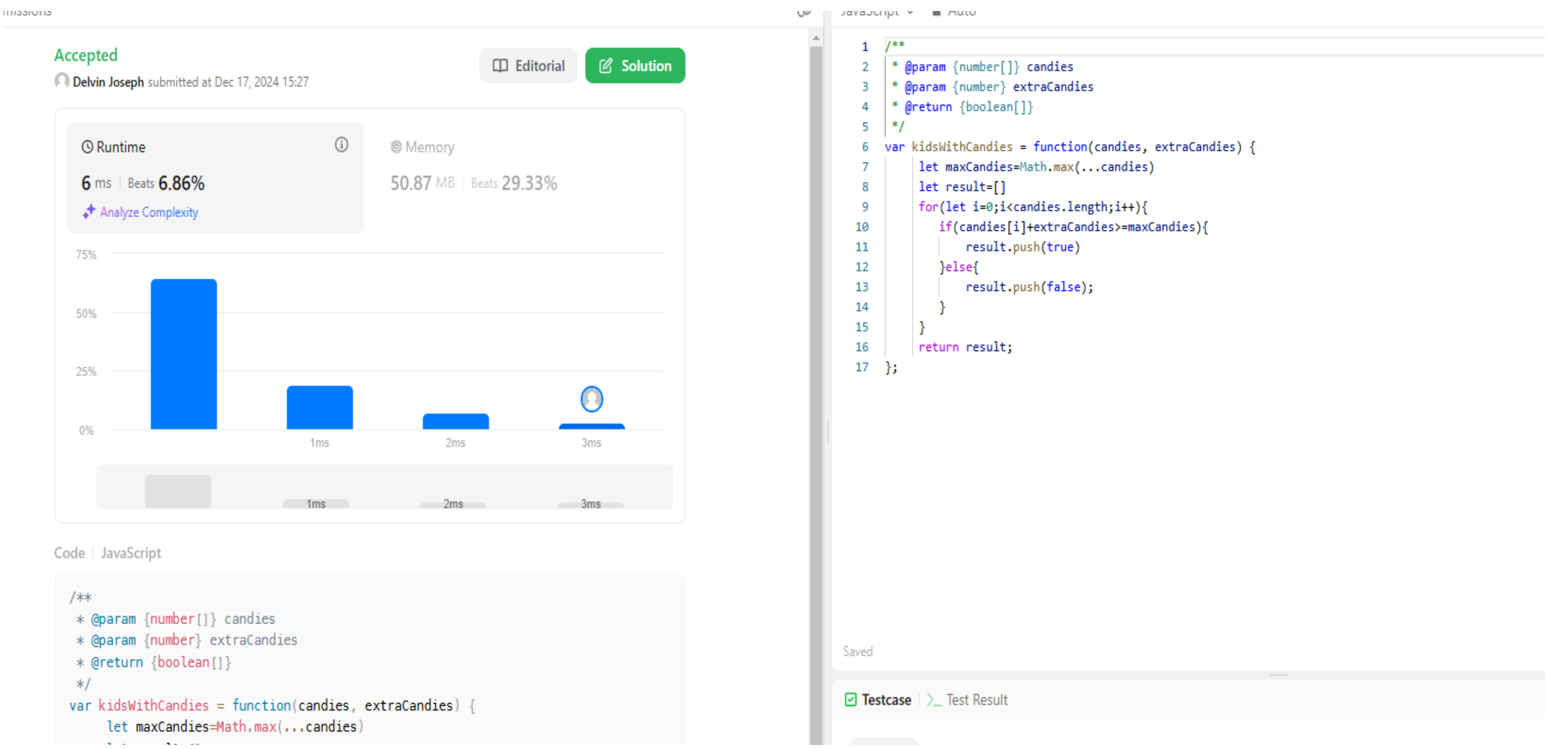
1) Question: https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/description/

Solution Link:- https://leetcode.com/problems/kids-with-the-greatest-number-of-candies/submissions/1481001659

Time Complexity: O(n)

Space Complexity: O(n)

Screenshot:



Accepted

Delvin Joseph submitted at Dec 17, 2024 15:27

6 ms | Beats 6.86%

50.87 MB | Beats 29.33%

```
/**
 * @param {number[]} candies
 * @param {number} extraCandies
 * @return {boolean[]}
 */
var kidsWithCandies = function(candies, extraCandies) {
    let maxCandies=Math.max(...candies)
    let result=[]
    for(let i=0;i<candies.length;i++){
       if(candies[i]+extraCandies>=maxCandies){
           result.push(true)
       }else{
           result.push(false);
       }
    }
    return result;
};
```

Description:

Time complexity:O(n)

Iterates all over the n sized array and there is only one loop and pushs the element to the result array

Space complexity: O(n)

The result array keep on change with respect to the size of the array

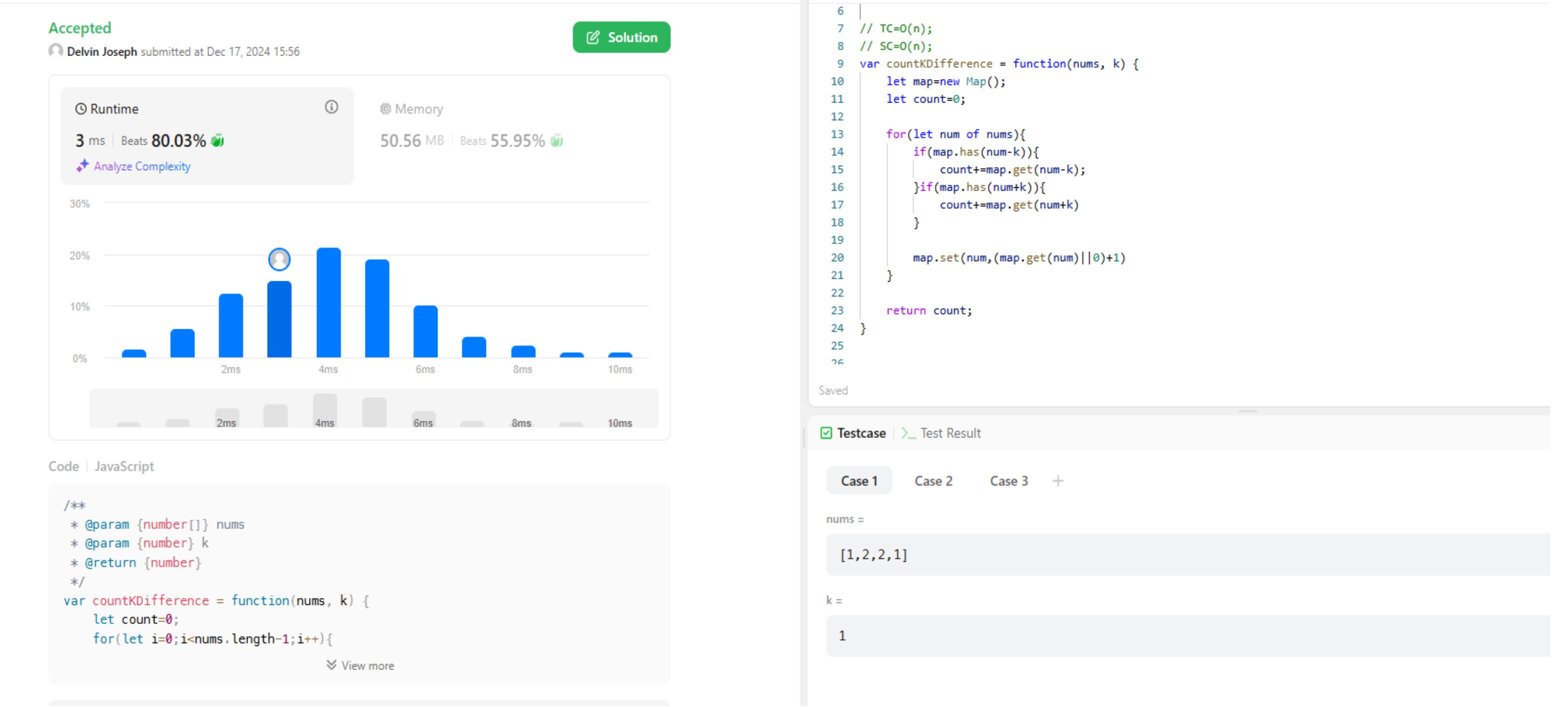
2) Question: [https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/description/](https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/description/)

Solution Link: [https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/submissions/1483452160](https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k/submissions/1483452160)

Time Complexity: O(n)

Space Complexity: O(n)

Screenshot:



Delvin Joseph submitted at Dec 17, 2024 15:56

3 ms | Beats 80.03%

50.56 MB | Beats 55.95%

Code | JavaScript

```
/**
 * @param {number[]} nums
 * @param {number} k
 * @return {number}
 */
var countKDifference = function(nums, k) {
    let count=0;
    for(let i=0;i<nums.length-1;i++){
```

View more

```

// TC=O(n);
// SC=O(n);
var countKDifference = function(nums, k) {
    let map=new Map();
    let count=0;

    for(let num of nums){
        if(map.has(num-k)){
            count+=map.get(num-k);
        }if(map.has(num+k)){
            count+=map.get(num+k)
        }

        map.set(num,(map.get(num)||0)+1)
    }

    return count;
}
```

nums =

[1,2,2,1]

k =

1

Description:

Time complexity:O(n)

Iterates all over the n sized array and there is only one loop

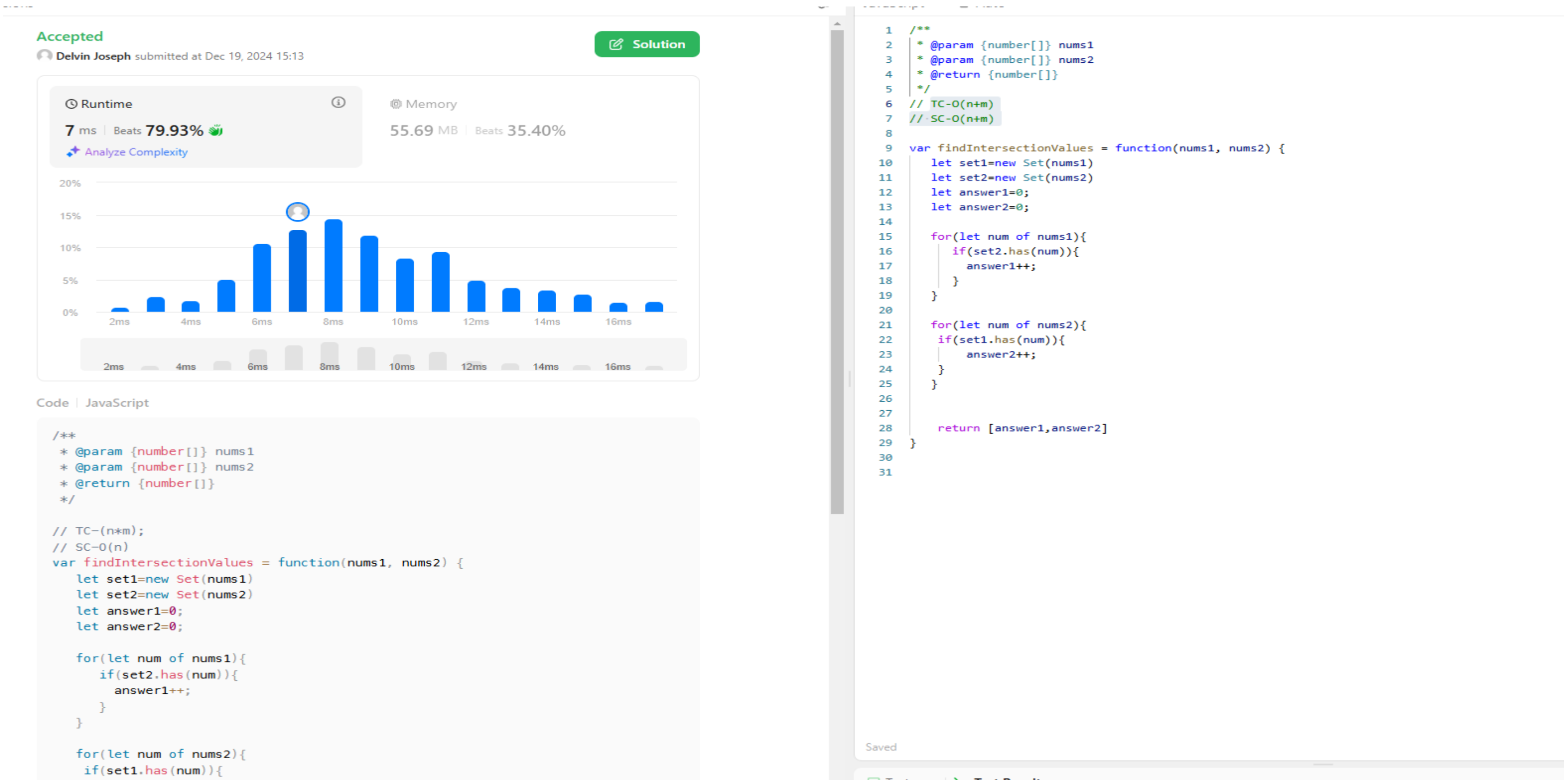
Space complexity: O(n)

Because of using the HashMap Data structure extra space is taking the space can be different but the worst case is O(n)

3)Question: [https://leetcode.com/problems/find-common-elements-between-two-arrays/description/](https://leetcode.com/problems/find-common-elements-between-two-arrays/description/)

Solution: [https://leetcode.com/problems/find-common-elements-between-two-arrays/submissions/1483455684](https://leetcode.com/problems/find-common-elements-between-two-arrays/submissions/1483455684)

Screenshot:



Accepted

Delvin Joseph submitted at Dec 19, 2024 15:13

7 ms | Beats 79.93%

55.69 MB | Beats 35.40%

```
/**
 * @param {number[]} nums1
 * @param {number[]} nums2
 * @return {number[]}
 */

// TC-(n*m);
// SC-O(n)
var findIntersectionValues = function(nums1, nums2) {
   let set1=new Set(nums1)
   let set2=new Set(nums2)
   let answer1=0;
   let answer2=0;

   for(let num of nums1){
      if(set2.has(num)){
        answer1++;
      }
   }

   for(let num of nums2){
    if(set1.has(num)){
```

```
/**
 * @param {number[]} nums1
 * @param {number[]} nums2
 * @return {number[]}
 */
// TC-O(n+m)
// SC-O(n+m)

var findIntersectionValues = function(nums1, nums2) {
   let set1=new Set(nums1)
   let set2=new Set(nums2)
   let answer1=0;
   let answer2=0;

   for(let num of nums1){
      if(set2.has(num)){
        answer1++;
      }
   }

   for(let num of nums2){
    if(set1.has(num)){
        answer2++;
    }
   }


    return [answer1,answer2]
}


```

Time Complexity: O(n+m)

Space Complexity: O(n+m)

Description:

Time Complexity: O(n+m)

It iterates 2 different  array with size of n and m there for the time takes to finish the loop is n+m

Space Complexity: O(n+m)

It takes 2 extra space which is set 1 and set 2 so it takes extra space with n+m complexity

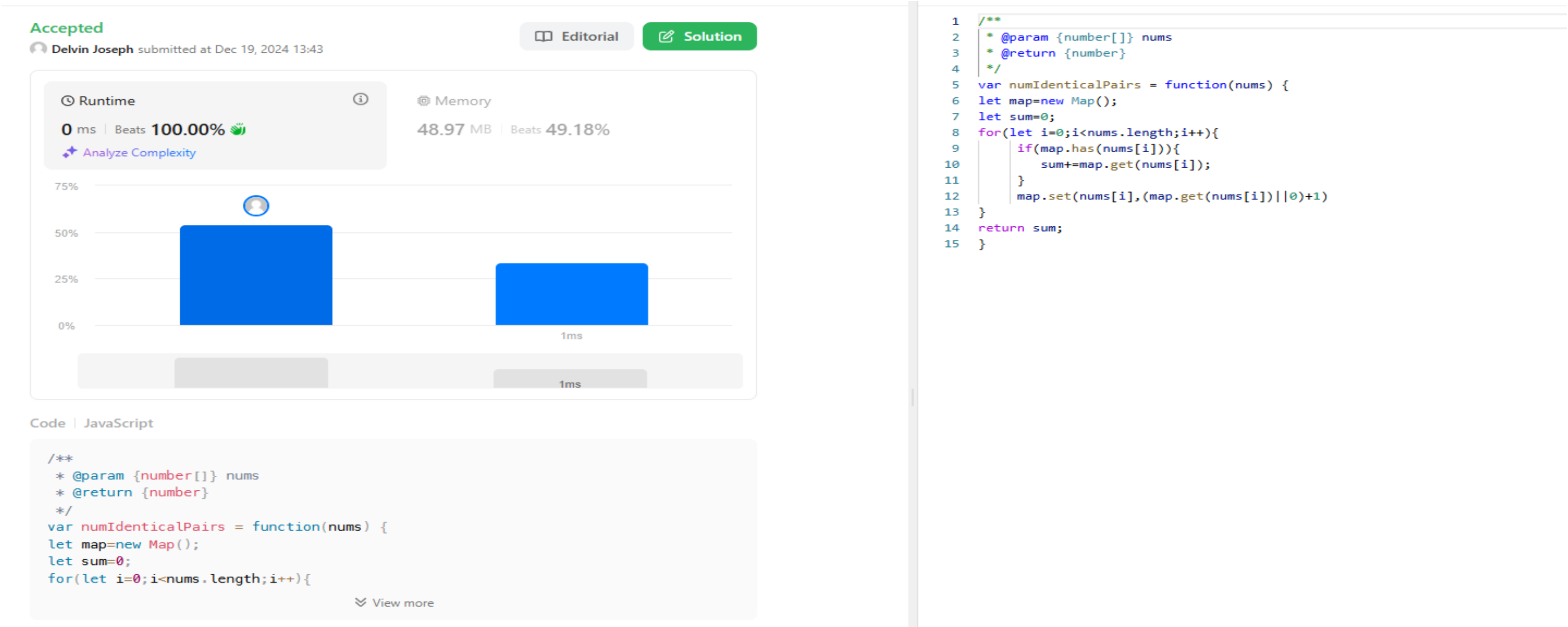
4)Question: https://leetcode.com/problems/number-of-good-pairs/description/

Solution: https://leetcode.com/problems/number-of-good-pairs/submissions/1482725397

Time Complexity: O(n)

Space Complexity: O(n)

Screenshot:



Accepted

Delvin Joseph submitted at Dec 19, 2024 13:43

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

48.97 MB | Beats 49.18%

Code | JavaScript

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var numIdenticalPairs = function(nums) {
let map=new Map();
let sum=0;
for(let i=0;i<nums.length;i++){
```

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var numIdenticalPairs = function(nums) {
let map=new Map();
let sum=0;
for(let i=0;i<nums.length;i++){
     if(map.has(nums[i])){
        sum+=map.get(nums[i]);
     }
     map.set(nums[i],(map.get(nums[i])||0)+1)
}
return sum;
}
```

Description:

Time Complexity:O(n)

It iterates all over the n –sized array there for the time complexity is O(n)

Space complexity is O(n) because the code takes extra space for the map Data structure

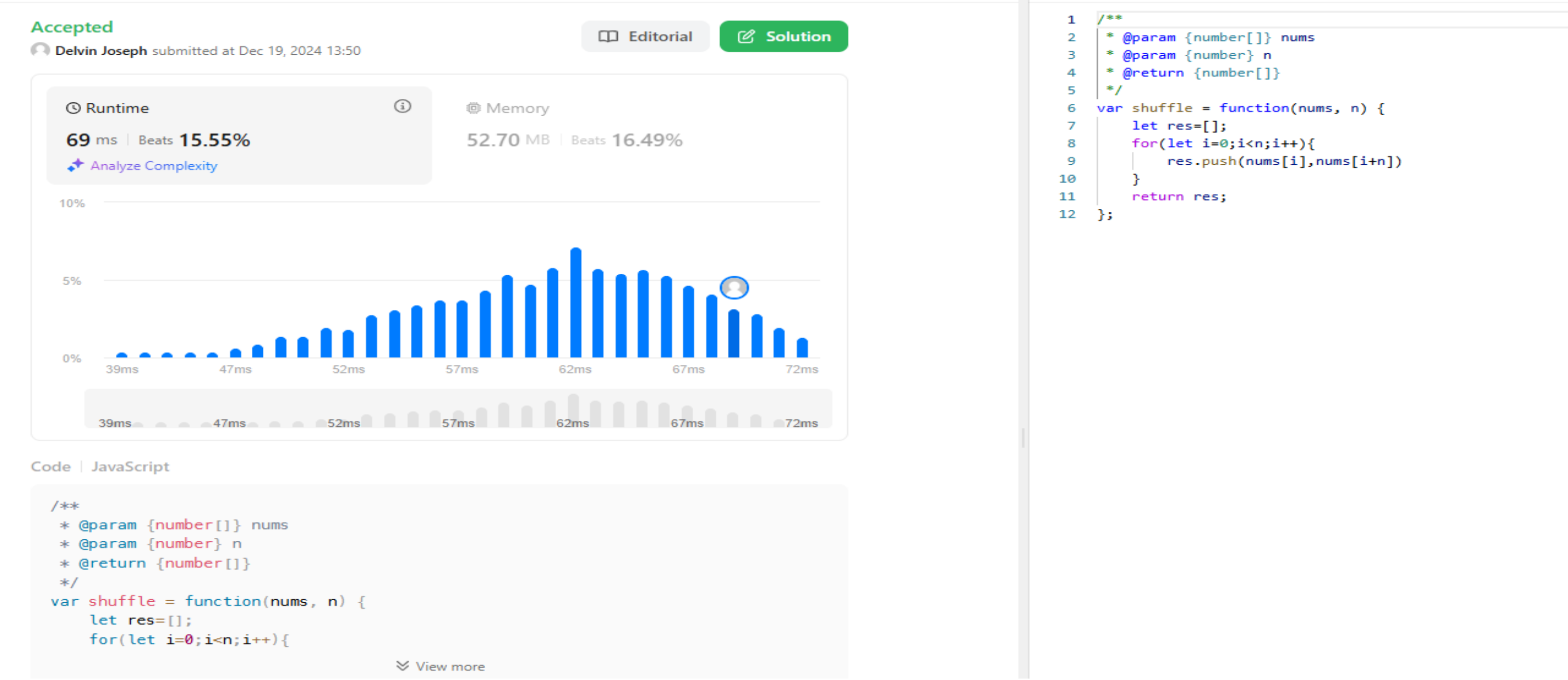
5)Question: [https://leetcode.com/problems/shuffle-the-array/description/](https://leetcode.com/problems/shuffle-the-array/description/)

Solution: [https://leetcode.com/problems/shuffle-the-array/submissions/1482730307](https://leetcode.com/problems/shuffle-the-array/submissions/1482730307)

Time Complexity: O(n)

Space Complexity: O(n)

Screenshot:



Accepted

Delvin Joseph submitted at Dec 19, 2024 13:50

Runtime

69 ms | Beats 15.55%

Memory

52.70 MB | Beats 16.49%

```
/**
 * @param {number[]} nums
 * @param {number} n
 * @return {number[]}
 */
var shuffle = function(nums, n) {
    let res=[];
    for(let i=0;i<n;i++){
```

```
/**
 * @param {number[]} nums
 * @param {number} n
 * @return {number[]}
 */
var shuffle = function(nums, n) {
    let res=[];
    for(let i=0;i<n;i++){
        res.push(nums[i],nums[i+n])
    }
    return res;
};
```

Description:

Time Complexity:O(n)

It iterates all over the n –sized array there for the time complexity is O(n)

Space complexity is O(n) because the code takes extra space for the result array it can get n space with respect to the size of the array