Assignment 7

Name : Hrishikesh Rajan
Email : hrishikeshrajan3@gmail.com

---

1) 1. Given an integer array nums of length n and an integer target, find three integers in nums such that the sum is closest to the target.[Amazon] You need to return the sum of three integers.

For example:arr = [-1, 2, 1, -4], target = 1
Output: 2
Explanation: [-1+2+1] = 2 (The sum that is closest to the target is 2)

ans)

CODE:

```
function closest(nums,target){

   let closest_sum =100000000;

   nums.sort((a, b) => a - b);

   for (let i = 0; i < nums.length - 2; i++){

       let pointer1 = i+1;
       let pointer2 = nums.length-1;

       while(pointer1<pointer2){


           let sum = nums[i]+nums[pointer1]+nums[pointer2];

           if(Math.abs(target -sum) < Math.abs(target - closest_sum) ){
               closest_sum = sum;
           }

           if(sum > target){
               pointer2--;
           }
```

```
            else{
                pointer1++
            }
        }

    }

    return closest_sum;

}
```

INPUT :

```
nums = [-1, 2, 1, -4];
const result = closest(nums,1);
console.log(result)
```

OUTPUT

```
2
```

CODE FLOW

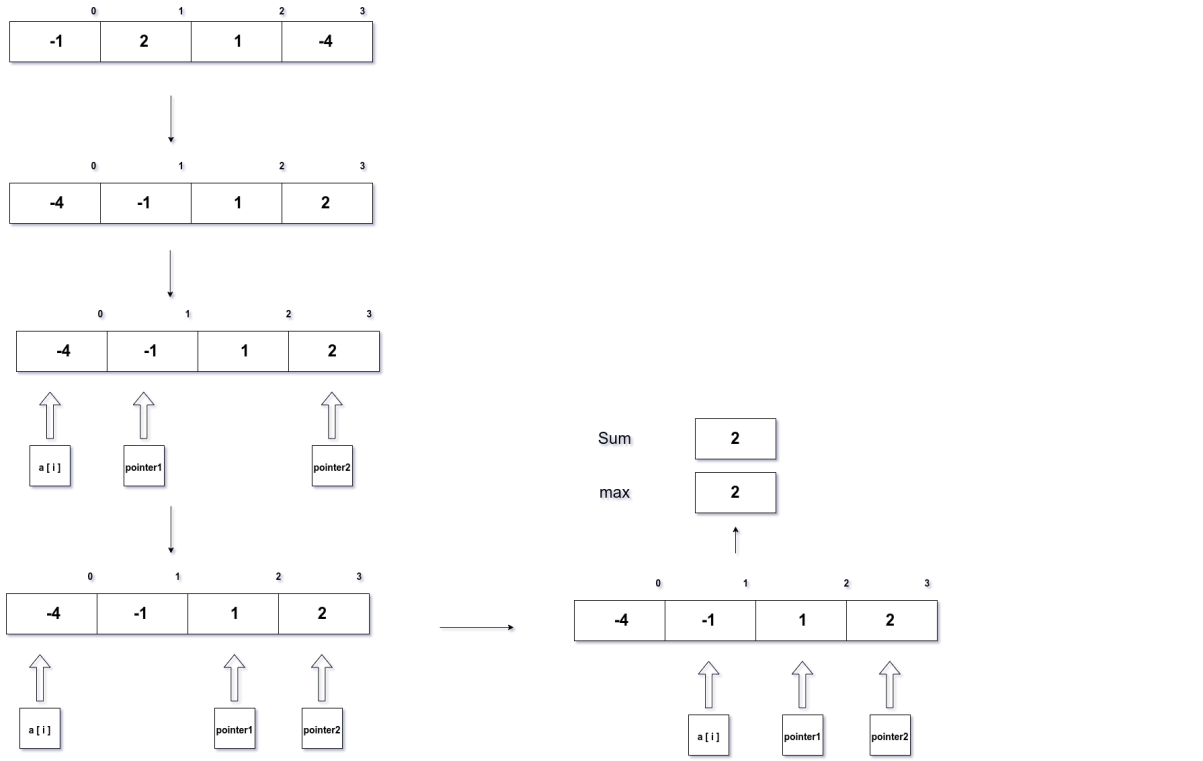

fig 1

ANALYSIS:

Here we use two steps, step 1 is to sort the input array and step two applying the calculations

In step 1 sorting , We use an inbuilt method to sort the array in ascending order.The Time complexity of inbuilt sort function can be vary from O (n ^2) to O (n * log (n)). Since the execution time complexity is added parallel to the time complexity of the sum calculation function is neglected .

In step 2 the calculator phase is done, for calculation we can use normal brute force approach which can cause O (n ^ 3), which is very very slow and expensive process.Hence we use a better approach called **Two Pointer Approach** (fig 1)

EXPLANATION:

When we talk about the time complexity of the total code

For the first step of the code the sorting takes O (n ^ 2) ‘this complexity can vary from language to language ’ .But here always considering the default one.
Time Complexity for sorting = O (n ^2)

Time Complexity for calculating the sum which takes O (n ^2).Since two point algorithm is famous for getting the summation in O (n), since here we have three variables need to be add hence we require a nested loop with each loop runs for n times approximately

Hence, the total time complexity is for this solution is = O (n ^ 2) + O (n ^ 2)
= O (n ^ 2)
Auxiliary Space Complexity = O (1), Since we are not using any extra array