

## Sum Pair closest to target

Difficulty: Easy

Accuracy: 44.75%

Submissions: 57K+

Points: 2

Given an array **arr[]** and a number **target**, find a pair of elements (a, b) in **arr[]**, where  $a \leq b$  whose sum is closest to **target**.

**Note:** Return the pair in sorted order and if there are multiple such pairs return the pair with maximum absolute difference. If no such pair exists return an empty array.

### Examples:

**Input:** arr[] = [10, 30, 20, 5], target = 25

**Output:** [5, 20]

**Explanation:** As  $5 + 20 = 25$  is closest to 25.

**Input:** arr[] = [5, 2, 7, 1, 4], target = 10

**Output:** [2, 7]

**Explanation:** As (4, 7) and (2, 7) both are closest to 10, but absolute difference of (2, 7) is 5 and (4, 7) is 3. Hence, [2, 7] has maximum absolute difference and closest to target.

**Input:** arr[] = [10], target = 10

```
1 // } Driver Code Ends
43 class Solution {
44     public List<Integer> sumClosest(int[] arr, int target) {
45         int n=arr.length;
46         Arrays.sort(arr);
47         int left=0;
48         int right=n-1;
49         int minDiff = Integer.MAX_VALUE;
50         List<Integer>Li=new ArrayList<>();
51         while(left<right){
52             int sum=arr[left]+arr[right];
53             int diff = Math.abs(target-sum);
54             if(diff<minDiff){
55                 minDiff=diff;
56                 Li=Arrays.asList(arr[left],arr[right]);
57             }if (sum>target){
58                 right--;
59             }else if(sum<target){
60                 left++;
61             }else{
62                 return Arrays.asList(arr[left],arr[right]);
63             }
64         }
65         return Li;
66     }
67 }
```

