**Two pointers**

1. Two sum sorted
2. Remove duplicates from sorted array
3. 3 Sum
4. 3 Sum closest
5. 3 Sum smaller

'''

Approach 1:

TC: O(N^2)

SC: O(N)

'''

import math

class Solution:

    def threeSumClosest(self, nums: List[int], target: int) -> int:

        nums.sort() # O(nlogn)

        res = nums[0] + nums[1] + nums[2]         # 1. initialize the result

        for i in range(len(nums)):

            left, right = i+1, len(nums)-1

            while left < right:

                s = nums[left] + nums[right] + nums[i]

                if s == target:# 2. If sum of 3 nums equals target return the sum

                    return s

                # 3. check the difference

                if abs(s-target) < abs(res - target):

                    res = s

                if s < target:        # 4. Increment left pointer

                    left += 1

                elif s > target:      # 5. Decrement right pointer

                    right -= 1

        return res

1. Squaring of a sorted array
2. Merge sorted array
3. Dutch national flag/Sort colors
4. 4sum
5. Trapping water
6. k-diff-pairs
7. Shortest unsorted continuous subarray
8. Interval list intersections
9. Reverse string
10. Reverse vowels of a string
11. Move zeros
12. Minimum size subarray sum
13. Candy crush
14. Median of two sorted arrays