3 Sum Closest

To solve the 3 Sum Closest problem, you can adapt the classic 3 Sum two pointer technique, but instead of finding a triplet that sums to exactly 0, you'll track the closest sum to the target seen so far.

Strategy Overview:

1. Sort the array nums.
2. Iterate through each element i:
   - use two pointers: left = i + 1, right = n - 1
   - for each triplet nums[i] + nums[left] + nums[right], check:
     - if sum == target: you've found the exact answer → return it
     - else, if abs(sum - target) < abs(closest - target), update closest
     - move left & right based on whether sum < target & sum > target
3. Return the best closest sum at the end.

Initialization: Use a large dummy value like:

```
int closest = nums[0] + nums[1] + nums[2];
```

Pointer Movement Logic:

- If sum < target → try larger sums → left++
- If sum > target → try smaller sums → right--

Time Complexity:

- Sorting: O(n log n)
- Two pointer loop: O(n²)
- Efficient enough for n ≤ 500