Two Pointers

Walk two indices through the same list, usually from both ends, so a comparison problem finishes in one pass instead of a nested loop.

WHEN TO REACH FOR IT

- You need a pair, or a small fixed group, of elements that satisfy some condition against a target.
- A brute-force answer nests two loops (O(n^2)), but the data is sorted or the order does not matter.
- You are checking whether a list or string reads the same from both directions, like a palindrome.
- You want O(1) extra space instead of a hash map or a second array.

1 SET UP THE WALK

```
# Two numbers in nums that sum to target.
def two_sum(nums, target):
    a = 0  # left pointer
    b = len(nums) - 1  # right pointer

```

2 WALK UNTIL THEY MEET

```
    while a < b:
        current_sum = nums[a] + nums[b]

```

3 DECIDE AND MOVE

```
        if current_sum == target:
            return [a, b]  # found it
        elif current_sum < target:
            a += 1  # raise the sum
        else:
            b -= 1  # lower the sum

```

4 ANSWER

```
    return None  # no pair found
```

Every problem here is the same walk. Step 3 says what you are looking for.

1. **Set up the walk.** Two pointers at opposite ends. Every problem here starts by placing them.
2. **Walk until they meet.** The loop never changes. Strict <, never <=.
3. **Decide and move.** Your problem lives here: the check, and which pointer moves.
4. **Answer.** Return the find, or the no-match answer.

COMPLEXITY

O(n) TIME **O(1)** SPACE

Sorting first (if the list is not already sorted) costs O(n log n). The pointer sweep itself is a single O(n) pass.

WATCH OUT

- This only works cleanly when the list is sorted, or order does not matter. Sort first if it is not.
- Every inner loop that nudges a pointer needs its own bound check, like left < right in Valid Palindrome, or you will walk off the end of the string.
- The loop condition is strict (<), not <=. Once the pointers meet or cross, there is nothing left to compare.

VARIATIONS

Pairs and triplets
Look for a fixed number of elements that hit a target sum, like Sum of Three or Best Time to Buy and Sell Stock.

Scan from both ends
Two pointers converge from opposite ends of the same string or list, like Valid Palindrome, Valid Palindrome II, or Trapping Rain Water.