Two Sum Problem - Python Solutions

Problem: Given an array of integers nums and an integer target,
determine if there are two numbers such that they add up to target.

1. HashMap / Set Approach

Steps you used to solve the problem:

- Create an empty set to store visited numbers.
- Iterate through the array, and for each number:
    - Check if target - num exists in the set.
    - If yes, return True because we found a pair.
    - Otherwise, add the number to the set.
- If no pair is found after the loop, return False.

```
class Solution:
    def twoSum(self, arr, target):
        seen = set()
        for num in arr:
            if target - num in seen:
                return True
            seen.add(num)
        return False
```

✔ Time Complexity: O(n)

✔ Space Complexity: O(n)

2. Two-Pointer Approach (After Sorting)

Steps you used to solve the problem:

- First sort the array.
- Initialize two pointers: i at start, j at end.
- While i < j:
    - If arr[i] + arr[j] > target, move j left (decrease sum).
    - If arr[i] + arr[j] < target, move i right (increase sum).
    - If equal, return True.
- If no pair is found, return False.

```
class Solution:
    def twoSum(self, arr, target):
        n = len(arr)
        arr.sort()
        i, j = 0, n - 1
        while i < j:
            if arr[i] + arr[j] > target:
                j -= 1
            elif arr[i] + arr[j] < target:
                i += 1
            else:
                return True
        return False
```

✔ Time Complexity: O(n log n) (due to sorting)

✔ Space Complexity: O(1)