2

Two Pointers Techniques

Recap:

- List Revision
- Handling multiple queries
- Counting number of operations
- Prefix sum technique

Problem 4:

Problem: Find Pair with Target Sum

Given a sorted array of integers and a target sum, determine if there exists a pair of numbers in the array whose sum equals the target.

Length of list <= 10^3

Brute Force Solution :

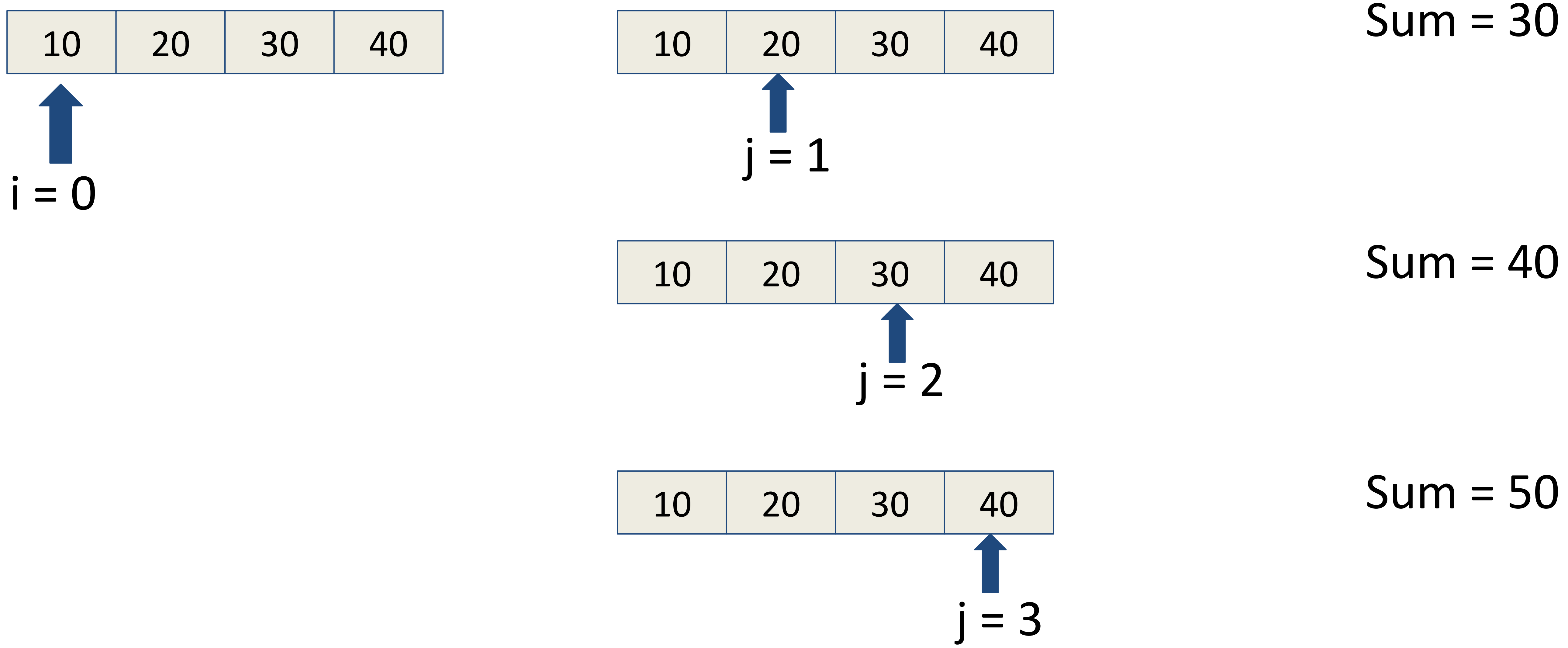
1. Use two nested loops to check all possible pairs of numbers in the array.
2. For each pair, calculate their sum.
3. If any pair's sum equals the target, return True.
4. If no such pair is found after checking all possibilities, return False.

Brute Force Dry Run:

Given List:

10	20	30	40

Target = 60

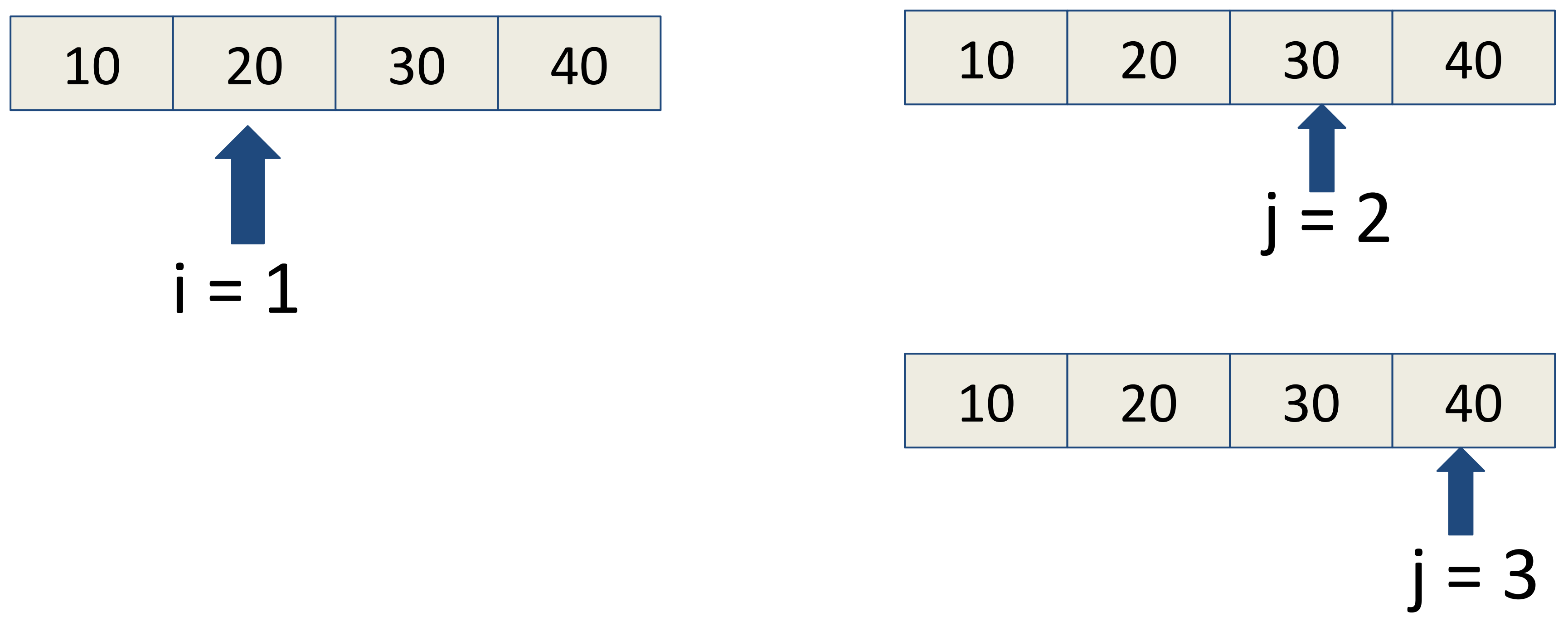


Brute Force Dry Run:

Given List:

10	20	30	40

Target = 60

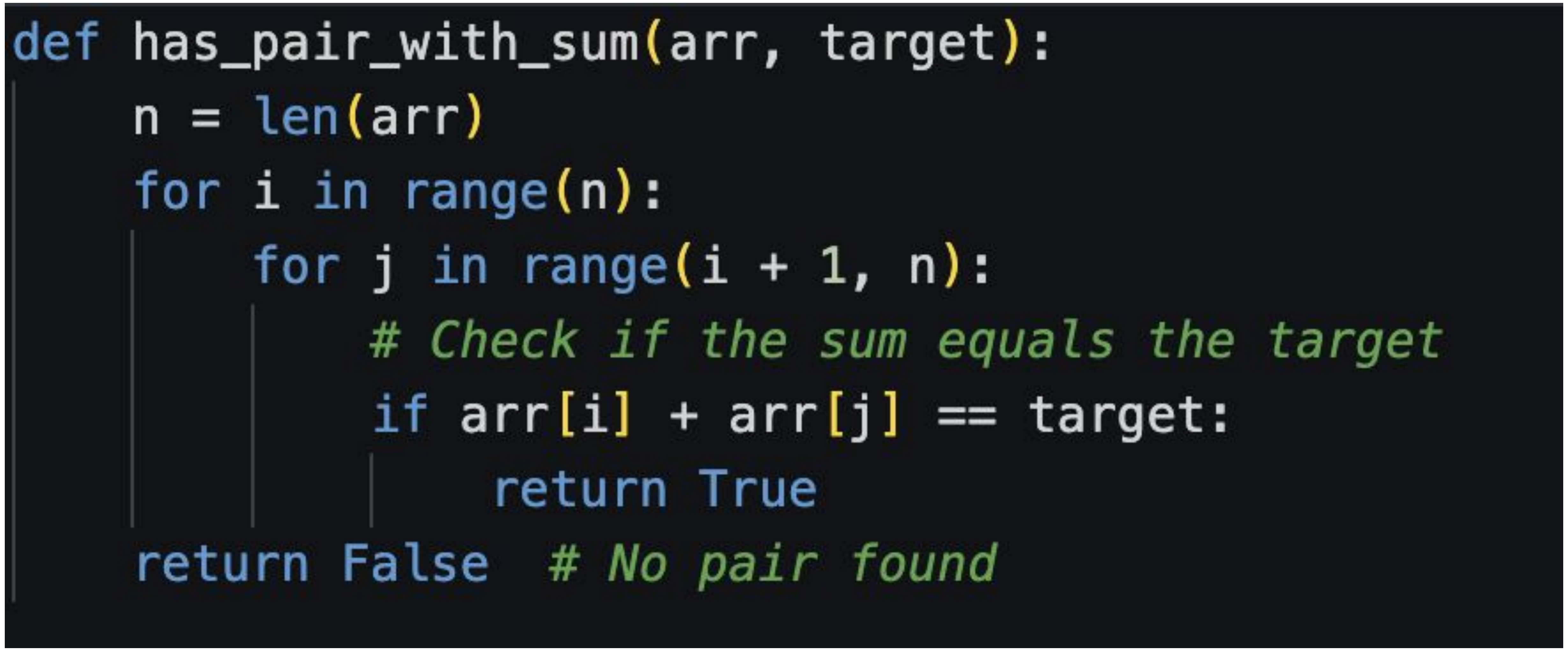


Sum = 50

Sum = 60

Found target. Return True

Brute Force Code:



```python
def has_pair_with_sum(arr, target):
    n = len(arr)
    for i in range(n):
        for j in range(i + 1, n):
            # Check if the sum equals the target
            if arr[i] + arr[j] == target:
                return True
    return False  # No pair found
```

What is the maximum number of operations?

Maximum number of operations:

- Outer loop runs n times with i going from 0 to n-1.
- Inner loop runs n–(i+1) times with j going from i+1 to n.
  - When i = 0 -> Inner loop runs n-1 times
  - When i = 1 -> Inner loop runs n-2 times
  - ……….
  - ……….
  - When i = n-2 -> Inner loop runs 1 times
  - When i = n-1 -> Inner loop doesn't run

Total number of operations = ((n-1) * n) / 2 = n C 2

Problem 5:

Problem: Find Pair with Target Sum

Given a sorted array of integers and a target sum, determine if there exists a pair of numbers in the array whose sum equals the target.

Length of list <= 10^5

Will brute force solution work?

Let’s calculate maximum number of operations:

- We are checking sum for every pair.
- Total number of pairs = n C 2
- Total number of operations: k * n C 2 where k is some constant
- k * 10^5 * (10^5-1) / 2 ~= 10^{10}
- Will definitely not work in time limit.

Can we do better?

Can we do better?

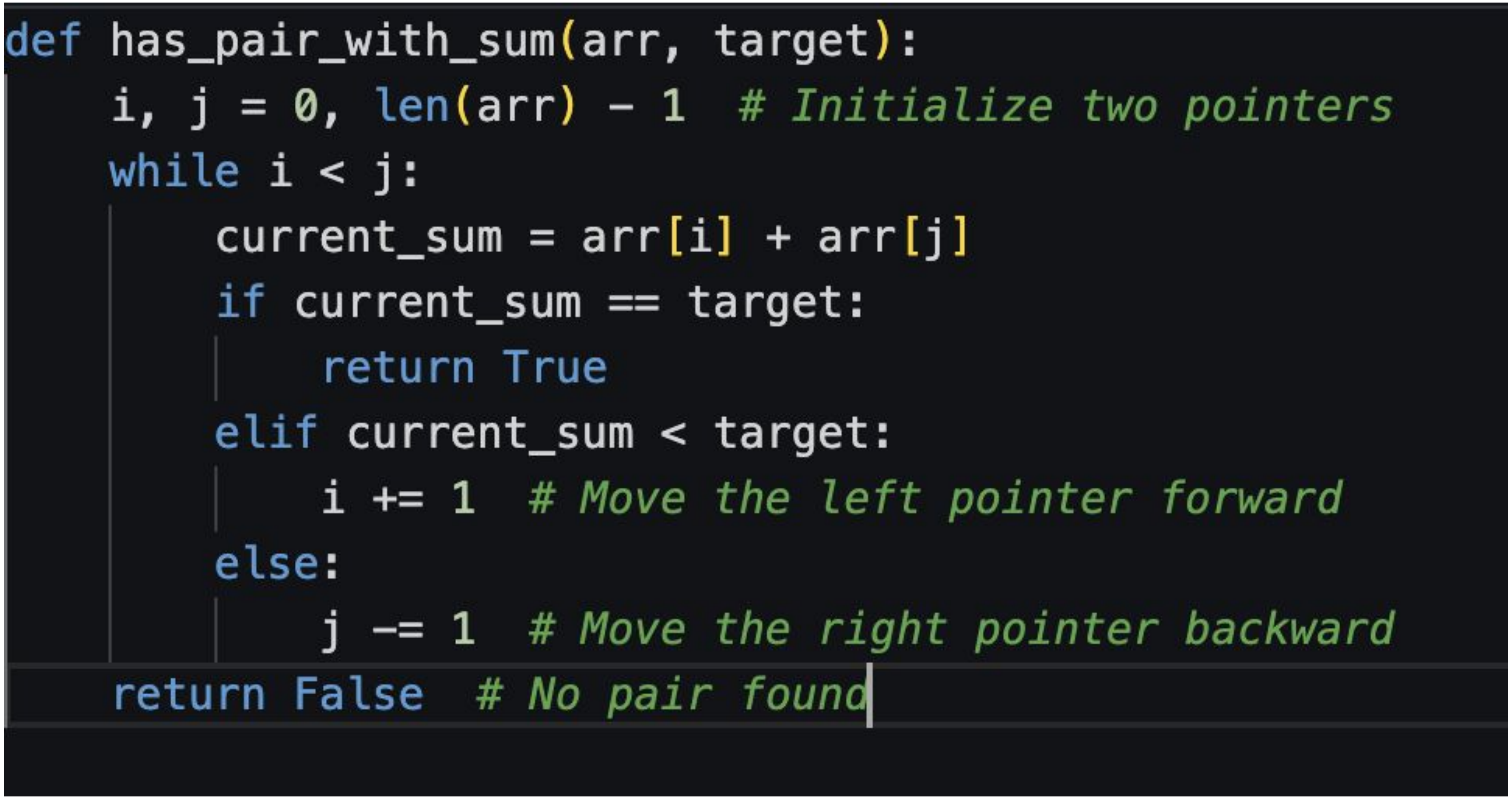
The given array is sorted. Can we use that to our advantage?

Can we do better?

The given array is sorted. Can we use that to our advantage?

- **If the sum of two numbers is less than the target, moving the smaller number to a larger one is the only option to increase the sum.**
- **If the sum is greater than the target, moving the larger number to a smaller one is the only option to decrease the sum.**
- **After every step, the search space gets reduced by 1.**

Optimal Code: Two Pointers



```python
def has_pair_with_sum(arr, target):
    i, j = 0, len(arr) - 1  # Initialize two pointers
    while i < j:
        current_sum = arr[i] + arr[j]
        if current_sum == target:
            return True
        elif current_sum < target:
            i += 1  # Move the left pointer forward
        else:
            j -= 1  # Move the right pointer backward
    return False  # No pair found
```

What is the maximum number of operations?

Maximum number of operations:

- Loop starts at i=0 and j = n-1 and runs till i < j.
- After every iteration either i increases or j decreases.
- Hence after n iterations i > j.
- Let’s assume every iteration takes k operations where k is some constant:

Total number of operations = k * n ~= n

Homework:

Prove the correctness of two pointers approach.

Summary:

- **List:**
  - **Iterating and Indexing,**
  - **Operations: Addition, Modification, Removal, Slicing**
  - **Inbuilt functions and methods.**
- **Prefix Sum Technique**
- **Two Pointers Technique**

Summary Quiz

2D List input and Indexing

Thank You!