Two Pointer Technique in TypeScript

What is Two Pointer Technique?

Two Pointer is an algorithmic technique used to reduce time complexity by using two indices (pointers) to traverse a data structure, usually an array or a string.

- It's not a Data Structure.
- It is used when you have to iterate over data from both ends or scan it in a linear pass with two indexes.

Common Use Cases

1. Finding target sum in a sorted array
2. Reversing an array or string
3. Removing duplicates from sorted array
4. Checking if a string is a palindrome
5. Trapping Rain Water
6. Container With Most Water

Example 1: Pair with Target Sum (Sorted Array)

```
function twoSumSorted(arr: number[], target: number): [number, number] | null {
  let left = 0;
  let right = arr.length - 1;

  while (left < right) {
    const sum = arr[left] + arr[right];
    if (sum === target) return [arr[left], arr[right]];
    else if (sum < target) left++;
    else right--;
  }
```

```
  return null;
}
```

Example 2: Is Palindrome

```
function isPalindrome(str: string): boolean {
  let left = 0;
  let right = str.length - 1;

  while (left < right) {
    if (str[left] !== str[right]) return false;
    left++;
    right--;
  }

  return true;
}
```

Example 3: Remove Duplicates from Sorted Array

```
function removeDuplicates(nums: number[]): number {
  if (nums.length === 0) return 0;

  let i = 0;
  for (let j = 1; j < nums.length; j++) {
    if (nums[i] !== nums[j]) {
      i++;
      nums[i] = nums[j];
    }
```

```
  }

  return i + 1;
}
```

Summary

Two Pointer = 2 indexes (e.g., left, right) that move across array/string to solve problems more efficiently.

Mostly used for linear problems where nested loops can be avoided.