DSA SERIES

- Learn Coding

Topic to be Covered today

Two pointers Approach

On String

LETS START TODAY'S LECTURE

Two pointers are like two soldiers who march from either:

- **Start and end** of the array (opposite directions), or
- **Start and next** element (same direction)

They help **efficiently search or solve problems** that involve:

- Sorted arrays
- Linked lists
- Strings
- Searching for pairs/subarrays, etc.

Types of Two Pointer Techniques

1.Opposite Direction

2. Same Direction

Problem : 344

Reverse String

Write a function that reverses a string.

The input string is given as an array of characters s.

You must do this by modifying the input array in-place with O(1) extra memory.

Example 1:

Input: s = ["h","e","l","l","o"]

Output: ["o","l","l","e","h"]

```
class Solution {
public:
    void reverseString(vector<char>& s) {
        int n = s.size();

        int i = 0, j = n - 1;

        while (i < j) {
            swap(s[i], s[j]);
            i++;
            j--;
        }
    }
};
```

Problem : 2000

Reverse Prefix of Word

Given a **0-indexed** string word and a character ch, **reverse** the segment of word that starts at index 0 and ends at the index of the **first occurrence** of ch (**inclusive**). If the character ch does not exist in word, do nothing.

- For example, if word = "abcdefd" and ch = "d",

then you should **reverse** the segment that starts at 0 and ends at 3 (**inclusive**).
The resulting string will be "dcbaefd".

Return *the resulting string*.

Input: word = "abcdefd", ch = "d" **Output:** "dcbaefd“
Explanation: The first occurrence of "d" is at index 3.
Reverse the part of word from 0 to 3 (inclusive), the resulting string is "dcbaefd".

```cpp
class Solution {
public:
    string reversePrefix(string word, char ch) {
        int n = word.size();

        // for (int i = 0; i < n; i++) {
        //     if (word[i] == ch) {
        //         idx = i;
        //         break;
        //     }
        // }

        int idx = word.find(ch);
        if(idx == string::npos){
            return word;
        }
```

```
            // if (idx == -1) {
            //     return word;
            // }

            int left = 0;
            int right = idx;

            while (left < right) {
                swap(word[left], word[right]);
                left++;
                right--;
            }

            return word;
        }
    };
```

Problem : 345

Reverse Vowels of a String

Given a string s, reverse only all the vowels in the string and return it.
The vowels are 'a', 'e', 'i', 'o', and 'u’,
and they can appear in both lower and upper cases, more than once.

Input: s = "IceCreAm"
Output: "AceCreIm"
Explanation:
The vowels in s are ['I', 'e', 'e', 'A']. On reversing the vowels, s becomes "AceCreIm".

```cpp
class Solution {
public:
    bool isVowel(char c) {
        char ch = tolower(c); // lower and upper case both are handled here

        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }

    string reverseVowels(string s) {
        int left = 0;
        int right = s.size() - 1;

        while (left < right) {

            while (left < right && !isVowel(s[left]))
                left++;
            while (left < right && !isVowel(s[right]))
                right--;

            swap(s[left], s[right]);
            left++;
            right--;
        }
        return s;
    }
};
```

Learn coding

THANK YOU