Merge Strings Alternatively (2 pointers)

0 1 2 3 4

str1= "Rohit";

str2 = "Yadav";

0 1 2 3 4

same length

ans = "RYoahdiatv"

0 1 2 3 4 5 6 7 8 9

ans = ""

ans = "R"

ans = "RY"

ans = "RYo"

⋮

space

i

0 1 2 3 4

str1= "Rohit";

str2 = "Yadav";

0 1 2 3 4

j

str1.charAt(i)

+

str2.charAt(j)

ans = RY

$$\begin{array}{lccccc} & 0 & 1 & 2 & 3 & 4 \\ str1 = & \text{"R} & o & h & i & t\text{"} \\ str2 = & \text{"Y} & a & d & a & v\text{"} \\ & 0 & 1 & 2 & 3 & 4 \end{array}$$

$i = 0,$

$ans \mathrel{+}= \underline{str1.charAt(i)} + \underline{str2.charAt(i)}$

code1

T.C = O(N)
S.C = O(1)

```
public static String mergeString(String str1, String str2) {
    int i = 0;
    int j = 0;
    String ans = "";
    while ( i < str1.length() && j < str2.length() ) {

        ans += str1.charAt(i);
        ans += str2.charAt(j);

        i++;
        j++;
    }
    return ans;
}
```

code2

T.C = O(N)
S.C = O(1)

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str1 = scn.nextLine();
    String str2 = scn.nextLine();
    System.out.println(mergeString(str1, str2));
}

public static String mergeString(String str1, String str2) {
    String ans = "";

    for (int i = 0; i < str1.length(); i++) {
        ans = ans + str1.charAt(i) + str2.charAt(i);
    }

    return ans;
}
```

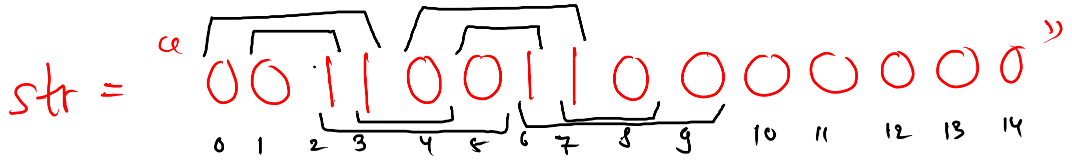
str1 = "Kunal"
0 1 2 3 4

str2 = "Anchi"
0 1 2 3 4

ans = "KAunncahli"

i=0, i=3
i=1, i=4
i=2,

Count Substring of 0 and 1



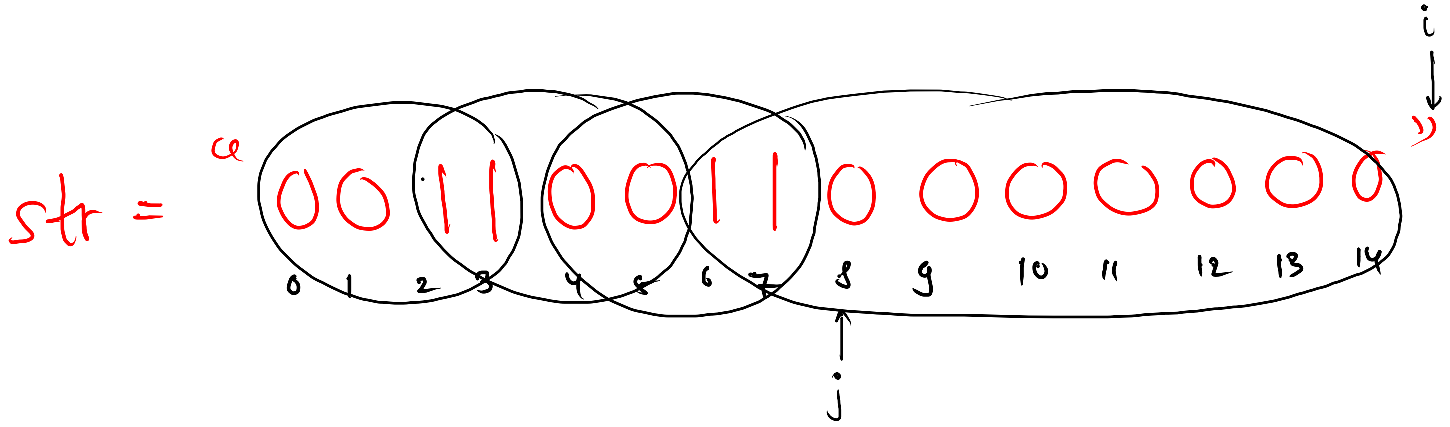
condition :-

↳ continuous substring where equal no. of 0's and equal no. of 1's should be grouped together

all substring :-

- (0,3) → 0011
- (1,2) → 01
- (2,5) → 1100
- (3,4) → 10
- (5,6) → 01
- (4,7) → 0011
- (7,8) → 10
- (6,9) → 1100

$\textcircled{8}$



countZero = ~~0~~ ~~2~~ 7

countOne = ~~0~~ 2

ans = 0 + 2 + 2 + 2 + 2 = 8

```java
public static int countSubstring(String str) {
    int n = str.length();
    int i = 0;
    int ans = 0;
    while ( i < n ) {
        int countZero = 0;
        int countOne = 0;
        if ( str.charAt(i) == '0' ) {
            while ( i < n && str.charAt(i) == '0' ) {
                countZero++;
                i++;
            }
            int j = i;
            while ( j < n && str.charAt(j) == '1' ) {
                countOne++;
                j++;
            }
        } else {
            while ( i < n && str.charAt(i) == '1' ) {
                countOne++;
                i++;
            }
            int j = i;
            while ( j < n && str.charAt(j) == '0' ) {
                countZero++;
                j++;
            }
        }
        ans = ans + Math.min( countZero, countOne );
    }
    return ans;
}
```

→ reseting

ans = 0+2+2+2

→ 6

str = " 1 1 1 0 0 1 1 0 0 0 0 0 "

j

i

6

	CountZero	CountOne
min=2	2	3
min=2	2	2
min=2	5	2

ans = 6

T.C = O(N)

operations = 2 x N

Long Pressed Name

```
        i
        ↓
str = "alex"

typed = "aaleex"
              ↑
              j
```

psudo code

logic

```
if (char at i == char at j){
        i++; j++;
}else {
        j++;
}
```

else if(char at j == char at j-1) False True

```
                  i
                  ↓
str =   " alex "

typed =" aaleeax "
              ↑
              j
```