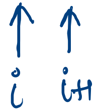
Power of a String

(Imp)

str = "aaabbbbcccdddddaaaaaa";

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21



ans = ~~0~~ ~~3~~ ~~4~~ ~~5~~ 7

len = ~~1~~ ~~2~~ ~~3~~
= ~~1~~ ~~2~~ ~~3~~ ~~4~~
= ~~1~~ ~~2~~ ~~3~~
= ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~
= ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7

psudo code

```
1) loop from 0 to (n-2)
    1.1) check if char at i
         and char at (i+1) are
         same
         then len++;
    1.2) not same
         then update ans
         for better len
         len = 1;
2) update ans if better len
```

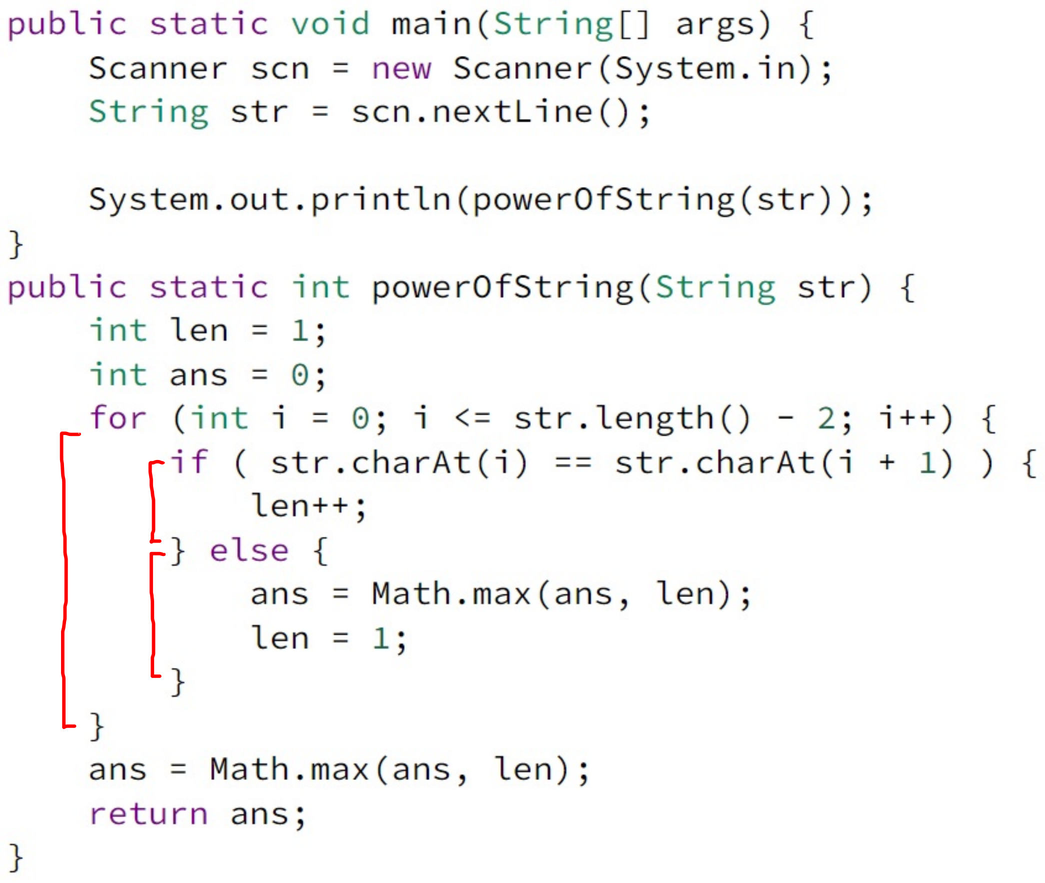
code

str = "aaabbbbcccdddddaaaaaaa";

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

↑

i



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();

    System.out.println(powerOfString(str));
}
public static int powerOfString(String str) {
    int len = 1;
    int ans = 0;
    for (int i = 0; i <= str.length() - 2; i++) {
        if ( str.charAt(i) == str.charAt(i + 1) ) {
            len++;
        } else {
            ans = Math.max(ans, len);
            len = 1;
        }
    }
    ans = Math.max(ans, len);
    return ans;
}
```

len = ~~5~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7

ans = ~~4~~ ~~5~~ 7

T.C = O(n)

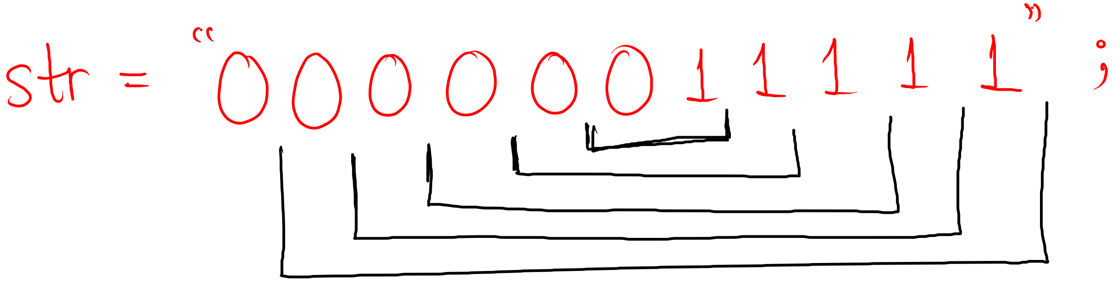
where, n is

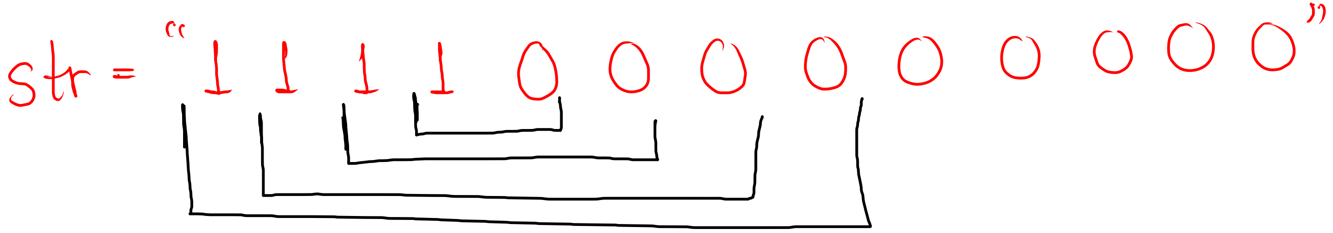
str.length

Count Substring of 0 and 1

- count substrings with
  - equal no. of 0's and 1's
  - all 1's and all 0's are together

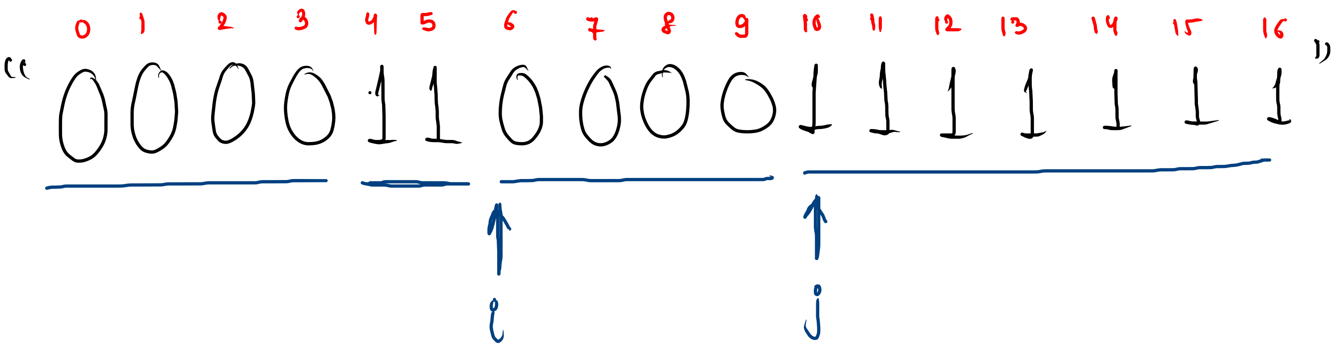
Ex:-


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


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

ans = min (no. of 0's , no. of 1's)

actual example



str = "00001100001111111"

no. of pairs = 3

↳ 0→5

↳ 4→9

↳ 6→16

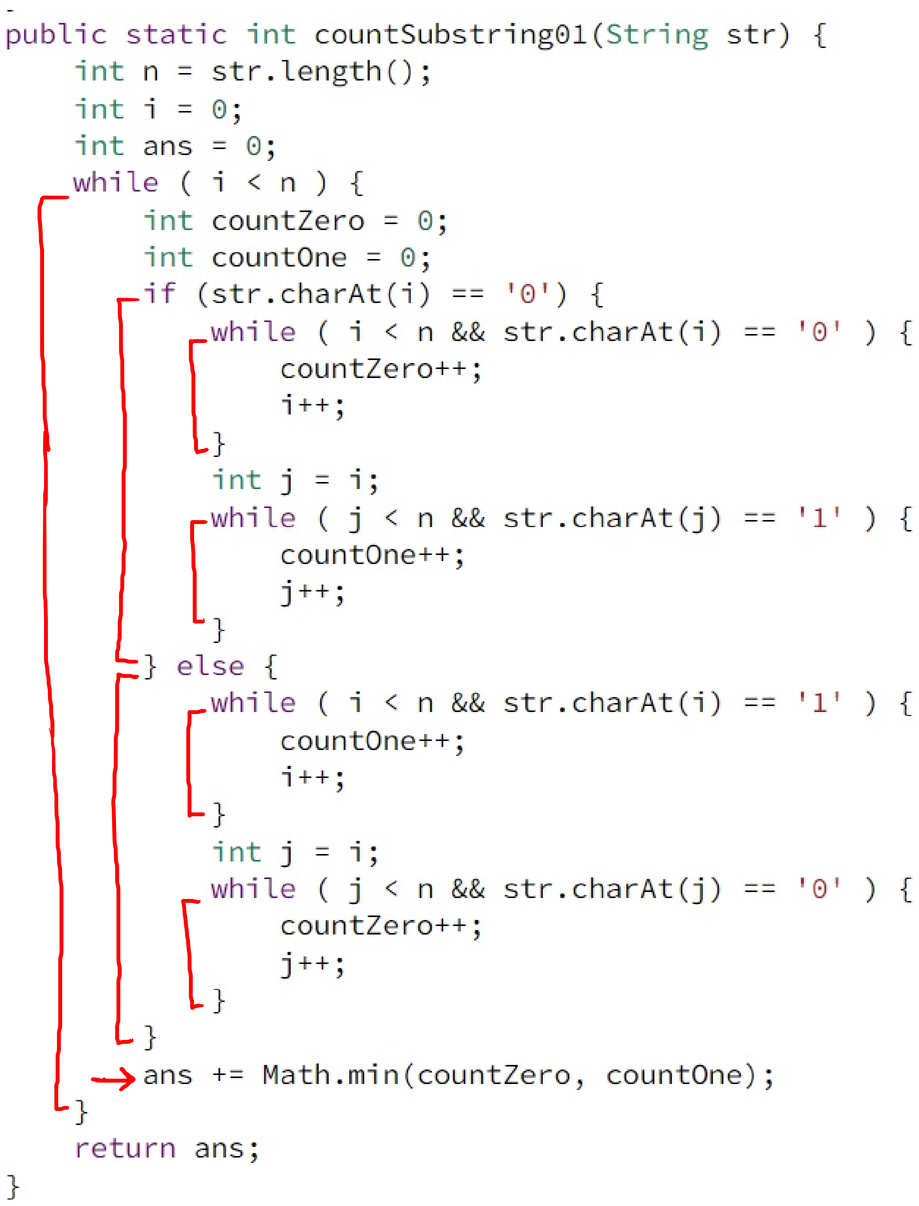
countZero = ~~4~~ ~~4~~ 4

countOne = ~~2~~ ~~2~~ 7

ans = 0 + 2 + 2 + 4

= 8

code



```java
public static int countSubstring01(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 += Math.min(countZero, countOne);
    }
    return ans;
}
```

dry run

actual example

$$\text{str} = \text{``}\begin{array}{ccccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\text{''}$$

↑↑
i j

ans = 0 + 2 + 2 + 4 + 0 = 8

countZero = ~~0~~ ~~4~~ ~~0~~ ~~4~~ ~~0~~ ~~4~~ 0

countone = ~~0~~ ~~2~~ ~~0~~ ~~2~~ ~~0~~ ~~7~~ ~~0~~ 7