Count Binary Substrings

Give a string `s`, count the number of non-empty (contiguous) substrings that have the same number of 0's and 1's, and all the 0's and all the 1's in these substrings are grouped consecutively.

Substrings that occur multiple times are counted the number of times they occur.

Example 1:

```
Input: "00110011"
Output: 6
Explanation: There are 6 substrings that have equal number of consecutive 1's and 0's: "0011", "01", "1100", "10", "0011", and "01".

Notice that some of these substrings repeat and are counted the number of times they occur.

Also, "00110011" is not a valid substring because all the 0's (and 1's) are not grouped together.
```

Example 2:

```
Input: "10101"
Output: 4
Explanation: There are 4 substrings: "10", "01", "10", "01" that have equal number of consecutive 1's and 0's.
```

Note:

- `s.length` will be between 1 and 50,000.
- `s` will only consist of "0" or "1" characters.

Solution 1

```
public int countBinarySubstrings(String s) {
    int prevRunLength = 0, curRunLength = 1, res = 0;
    for (int i=1;i<s.length();i++) {
        if (s.charAt(i) == s.charAt(i-1)) curRunLength++;
        else {
            prevRunLength = curRunLength;
            curRunLength = 1;
        }
        if (prevRunLength >= curRunLength) res++;
    }
    return res;
}
```

written by compton_scatter original link here

Solution 2

First, I count the number of 1 or 0 grouped consecutively.
For example "0110001111" will be `[1, 2, 3, 4]`.

Second, for any possible substrings with 1 and 0 grouped consecutively, the number of valid substring will be the minimum number of 0 and 1.
For example "0001111", will be `min(3, 4) = 3`, (`"01", "0011", "000111"`)

```
def countBinarySubstrings(self, s):
        s = map(len, s.replace('01', '0 1').replace('10', '1 0').split())
        return sum(min(a, b) for a, b in zip(s, s[1:]))
```

written by lee215 original link here

Solution 3

```
    def countBinarySubstrings(self, s):
        """
        :type s: str
        :rtype: int
        """
        n=len(s)
        res=0
        start=0
        lastcount=1
        for i in xrange(1,n):
            if s[i]!=s[i-1]:
                res+=1
                lastcount=i-start
                start=i
            else:
                if i-start<lastcount:
                    res+=1
        return res
```

written by yeyumujiang original link here

From Leetcoder.