Special String Again ★

Problem Submissions Leaderboard Editorial

A string is said to be a special string if either of two conditions is met:

- All of the characters are the same, e.g. aaa.
- All characters except the middle one are the same, e.g. aadaa.

A special substring is any substring of a string which meets one of those criteria. Given a string, determine how many special substrings can be formed from it.

For example, given the string $s = \texttt{mnonopoo}$, we have the following special substrings: $\{\texttt{m, n, o, n, o, p, o, o, non, ono, opo, oo}\}$.

Function Description

Complete the substrCount function in the editor below. It should return an integer representing the number of special substrings that can be formed from the given string.

substrCount has the following parameter(s):

- n: an integer, the length of string s
- s: a string

Input Format

The first line contains an integer, n, the length of s.

The second line contains the string s.

Constraints

$1 \leq n \leq 10^6$

Each character of the string is a lowercase alphabet, $\texttt{ascii[a-z]}$.

Output Format

Print a single line containing the count of total special substrings.

Sample Input 0

```
5
asasd
```

Sample Output 0

```
7
```

Explanation 0

The special palindromic substrings of $s = \texttt{asasd}$ are $\{\texttt{a, s, a, s, d, asa, sas}\}$

Sample Input 1

```
7
abcbaba
```

Sample Output 1

```
10
```

Explanation 1

The special palindromic substrings of $s = \texttt{abcbaba}$ are $\{\texttt{a, b, c, b, a, b, a, bcb, bab, aba}\}$

Sample Input 2

```
4
aaaa
```

Sample Output 2

```
10
```

Explanation 2

The special palindromic substrings of $s = \texttt{aaaa}$ are $\{\texttt{a}, \texttt{a}, \texttt{a}, \texttt{a}, \texttt{aa}, \texttt{aa}, \texttt{aa}, \texttt{aaa}, \texttt{aaa}, \texttt{aaaa}\}$

Change Theme JavaScript (Node.js)

```
26
27  // Complete the substrCount function below.
28  function substrCount(n, s) {
29      let count = n;
30      for (let i = 0; i < s.length; i++) {
31          let nextIndex = i;
32          while (s[i] === s[nextIndex + 1]) nextIndex++;
33
34          if (i !== nextIndex) {
35              const length = nextIndex - i;
36              count = count + (length * (length + 1)) / 2;
37              i = nextIndex;
38          } else {
39              let step = 1;
40              while (s[i + step] === s[i - step] && s[i + step] === s[i + 1]) {
41                  step++;
42                  count++;
43              }
44          }
45      }
46      return count;
47
48  }
49
```

Line: 46 Col: 18

Upload Code as File Test against custom input Run Code Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Sample Test case 2

Input (stdin) Download

```
5
asasd{-truncated-}
```

Download to view the full testcase

Your Output (stdout)

```
7
```

Expected Output Download

```
1 7
```