Alternating Characters

Isaac DeJager

March 15, 2023

Click here to view the HackerRank page

1 Problem Description

You are given a string containing characters A and B only. Your task it to change it into a string such that there are no matching adjacent characters. To do this, you are allowed to delete zero or more characters in the string. Your task is to find the minimum number of required deletions.

Example
$s = AABAAB$
Remove an A at positions 0 and 3 to make $s = ABAB$ in 2 deletions.

Function Description
Complete the alternatingCharacters function which has the following parameter:

- string s: a string

Returns

- int: the minimum number of deletions required

Constraints

- $1 \leq$length of $s \leq 10^5$
- s will consist only of characters A and B.

Sample Input
AAAA
BBBBB
ABABABAB
BABABA
AAABBB

Sample Output 3
4
0
0
4

Explanation

The characters marked red are the ones that can be deleted so that the string does not have matching adjacent characters.

AAAA → (3 deletions)
BBBBB → (4 deletions)
ABABABAB → ABABABAB (0 deletions)
BABABA → BABABA (0 deletions)
AAABBB → AB (4 deletions)

2 Solution

```
int alternatingCharacters(string s) {
    int x = 0;
    int y = 0;

    for (int i = 1; i < s.length(); i++) {
        if (s.at(i) == s.at(x)) {
            y++;
        } else {
            x = i;
        }
    }

    return y;
}
```

3 Analysis

The algorithm begins by creating two integer variables, both initialized at 0: x and y. x is an index counter that will be updated based on certain circumstances that will be described later. y is the total number of deletions and what will be returned. Creating variables is done in $2 * 0(1)$ time.

Then we enter a for loop that loops through each character in the string s starting at index $i = 1$ rather that $i = 0$. The contents of the for loop are simple. First, it runs an if statement that returns true if the character at index i matches the character at index x. If it does, y is incremented, because the character at i must be deleted. If it does not, then x is set equal to i. This algorithm runs in $O(n-1)$ time where n is the length of string s. It is optimized so that the string s is not changed at all, it is just the integer variables x, y, and i that change through the course of the algorithm.

The following example explains the logic.

Consider the string $s = AAABBBAABA$. The for loop runs as follows:

$i = 1$, $x = 0$, $y = 0$
$s.at(x) = A = A = s.at(i) \rightarrow s.at(1)$ is deleted, $y{+}{+}$
$s = AAABBBAABA$

$i = 2$, $x = 0$, $y = 1$
$s.at(x) = A = A = s.at(i) \rightarrow s.at(2)$ is deleted, $y{+}{+}$
$s = AAABBBAABA$

$i = 3$, $x = 0$, $y = 2$
$s.at(x) = A \neq B = s.at(i) \rightarrow x = i$
$s = AAABBBAABA$

$i = 4$, $x = 3$, $y = 2$
$s.at(x) = B = B = s.at(i) \rightarrow s.at(4)$ is deleted, $y{+}{+}$
$s = AAABBBAABA$

$i = 5$, $x = 3$, $y = 3$
$s.at(x) = B = B = s.at(i) \rightarrow s.at(5)$ is deleted, $y{+}{+}$
$s = AAABBBAABA$

$i = 6$, $x = 3$, $y = 4$
$s.at(x) = B \neq A\ s.at(i) \rightarrow x = i$
$s = AAABBBAABA$

$i = 7$, $x = 6$, $y = 4$
$s.at(x) = A = A = s.at(i) \rightarrow s.at(7)$ is deleted, $y{+}{+}$
$s = AAABBBAABA$

$i = 8$, $x = 6$, $y = 5$
$s.at(x) = A \neq B\ s.at(i) \rightarrow x = i$
$s = AAABBBAABA$

$i = 9$, $x = 8$, $y = 5$
$s.at(x) = A \neq B$ $s.at(i) \rightarrow x = i$
$s = AAABBBAABA$

With the 5 red characters deleted, $s = ABABA$ and y is returned. The reason for not repeatedly incrementing x like i is so that all characters between indexes x and i can be treated as though they are deleted. And as long as y is being tracked properly, we do not need to remember the indices of all deleted characters before index x.

The total time of this algorithm is:

$$2 * O(1) + O(n - 1) = O(n)$$

The space required is also minimal. Besides the passed string parameter s, only 2 integer variables are created.