ALTERNATING CHARACTERS

GIVEN: String = AABAAB ; where String ∈ {A, B}
And it cannot have matching adjacent characters
And you can delete 0 or more characters

FIND: The minimum number of character deletions so that the String has no matching adjacent (Side-by-Side) characters

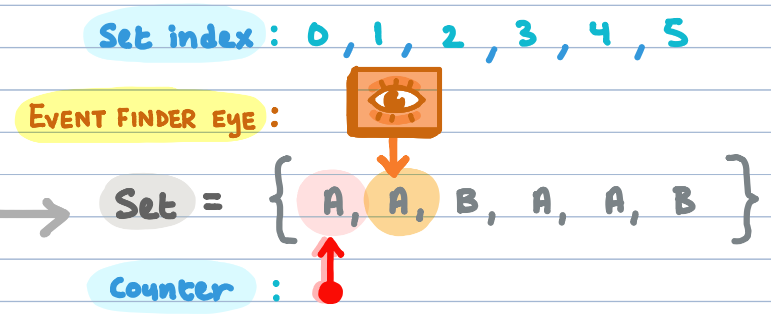
CONSTRAINTS :
$1 \le$ String queries ≤ 10
$1 \le |$ string length $| \le 10^5$

VISUALIZE PROBLEM

↓

MATHEMATICAL MODEL :

*(INTERVAL PROBLEM)



last_char_to_check = [A]

*where the Counter increments Starting @ index 0.
And the Event Finder waits for Counter to get to index 1 and then begins to checks for an Event as it increments and "Handles" it, Starting @ index 1, Just as it's stated below.
last_char_to_check holds the previous characters value.

DOMAIN OF DISCOURSE :

1D Linear Traversal (of a Set — "universe") of ∀ ∈ Set

EVENT BEING FOUND :

Checking if **Previous** char is Similar to the **Current** one

EVENT HANDLER :

Count the found comparison as a "deletion".
(i.e. the letters that repeat while adjacent to another letter that's the Same)

ALGORITHM: Mathematical Model Explanation

- First, grab the char at index 0 and Store it into a char, last_char_to_check

- Begin traversing the rest of the String set and compare the current char to last_char_to_check.

  As you move to a new index, update last_char_to_check with the current char.

  IFF the current char is the same as last_char_to_check then count it as a "deletion" before you update last_char_to_check with the current char
  Else, disregard the current char and keep traversing.

- Once you're done traversing return the count of the number of "deletions" found