100152. Remove Adjacent Almost-Equal Characters

My Submissions (/contest/biweekly-contest-119/problems/remove-adjacent-almost-equal-characters/submissions/)

Back to Contest (/contest/biweekly-contest-119/)

You are given a **0-indexed** string `word`.

In one operation, you can pick any index `i` of `word` and change `word[i]` to any lowercase English letter.

Return *the **minimum** number of operations needed to remove all adjacent **almost-equal** characters from* `word`.

Two characters `a` and `b` are **almost-equal** if `a == b` or `a` and `b` are adjacent in the alphabet.

User Accepted:	7248
User Tried:	8686
Total Accepted:	7509
Total Submissions:	16246
Difficulty:	Medium

Example 1:

```
Input: word = "aaaaa"
Output: 2
Explanation: We can change word into "acaca" which does not have any adjacent almost-equal characters.
It can be shown that the minimum number of operations needed to remove all adjacent almost-equal characters from wo
```

Example 2:

```
Input: word = "abddez"
Output: 2
Explanation: We can change word into "ybdoez" which does not have any adjacent almost-equal characters.
It can be shown that the minimum number of operations needed to remove all adjacent almost-equal characters from wo
```

Example 3:

```
Input: word = "zyxyxyz"
Output: 3
Explanation: We can change word into "zaxaxaz" which does not have any adjacent almost-equal characters.
It can be shown that the minimum number of operations needed to remove all adjacent almost-equal characters from wo
```

Constraints:

- `1 <= word.length <= 100`
- `word` consists only of lowercase English letters.

Java

```
class Solution {
    public int removeAlmostEqualCharacters(String word) {

    }
}
```