# Problem 1653: Minimum Deletions to Make String Balanced

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 65.60%
**Paid Only:** No
**Tags:** String, Dynamic Programming, Stack

## Problem Description

You are given a string `s` consisting only of characters `'a'` and `'b'`■■■■.

You can delete any number of characters in `s` to make `s` **balanced**. `s` is **balanced** if there is no pair of indices `(i,j)` such that `i < j` and `s[i] = 'b'` and `s[j]= 'a'`.

Return _the**minimum** number of deletions needed to make _`s` _**balanced**_.

**Example 1:**

**Input:** s = "aababbab" **Output:** 2 **Explanation:** You can either: Delete the characters at 0-indexed positions 2 and 6 ("aa _b_ abb _a_ b" -> "aaabbb"), or Delete the characters at 0-indexed positions 3 and 6 ("aab _a_ bb _a_ b" -> "aabbbb").

**Example 2:**

**Input:** s = "bbaaaaabb" **Output:** 2 **Explanation:** The only solution is to delete the first two characters.

**Constraints:**

* `1 <= s.length <= 105` * `s[i]` is `'a'` or `'b'`■■.

## Code Snippets

### C++:

```
class Solution {
public:
int minimumDeletions(string s) {

}
};
```

### Java:

```
class Solution {
public int minimumDeletions(String s) {

}
}
```

### Python3:

```
class Solution:
def minimumDeletions(self, s: str) -> int:
```