Problem 420: Strong Password Checker

Problem Information

Difficulty: Hard
Acceptance Rate: 15.19%
Paid Only: No
Tags: String, Greedy, Heap (Priority Queue)

Problem Description

A password is considered strong if the below conditions are all met:

* It has at least `6` characters and at most `20` characters. * It contains at least **one lowercase** letter, at least **one uppercase** letter, and at least **one digit**. * It does not contain three repeating characters in a row (i.e., `"B _**aaa**_ bb0"` is weak, but `"B** _aa_** b _**a**_ 0"` is strong).

Given a string `password`, return _the minimum number of steps required to make`password` strong. if `password` is already strong, return `0`._

In one step, you can:

* Insert one character to `password`, * Delete one character from `password`, or * Replace one character of `password` with another character.

Example 1:

Input: password = "a" **Output:** 5

Example 2:

Input: password = "aA1" **Output:** 3

Example 3:

Input: password = "1337C0d3" **Output:** 0

Constraints:

* `1 <= password.length <= 50` * `password` consists of letters, digits, dot `'.'` or exclamation mark `'!'`.

Code Snippets

C++:

```
class Solution {
public:
int strongPasswordChecker(string password) {

}
};
```

Java:

```
class Solution {
public int strongPasswordChecker(String password) {

}
}
```

Python3:

```
class Solution:
def strongPasswordChecker(self, password: str) -> int:
```