6095. Strong Password Checker II

My Submissions (/contest/biweekly-contest-80/problems/strong-password-checker-ii/submissions/) Back to Contest (/contest/biweekly-contest-80/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

A password is said to be **strong** if it satisfies all the following criteria:

- It has at least `8` characters.
- It contains at least **one lowercase** letter.
- It contains at least **one uppercase** letter.
- It contains at least **one digit**.
- It contains at least **one special character**. The special characters are the characters in the following string: `"!@#$%^&*()-+"`.
- It does **not** contain `2` of the same character in adjacent positions (i.e., `"aab"` violates this condition, but `"aba"` does not).

Given a string `password`, return `true` *if it is a* ***strong*** *password*. Otherwise, return `false`.

Example 1:

```
Input: password = "IloveLe3tcode!"
Output: true
Explanation: The password meets all the requirements. Therefore, we return true.
```

Example 2:

```
Input: password = "Me+You--IsMyDream"
Output: false
Explanation: The password does not contain a digit and also contains 2 of the same character in adjacent positions. Therefore,
```

Example 3:

```
Input: password = "1aB!"
Output: false
Explanation: The password does not meet the length requirement. Therefore, we return false.
```

Constraints:

- `1 <= password.length <= 100`
- `password` consists of letters, digits, and special characters: `"!@#$%^&*()-+"`.

JavaScript

```
const ord = (c) => c.charCodeAt();
const isLowerCase = (c) => { let x = ord(c); return x >= 97 && x <= 122; };
const isUpperCase = (c) => { let x = ord(c); return x >= 65 && x <= 90; };
const isDigit = (c) => '0123456789'.indexOf(c) != -1;
const isSpecial = (c) => '!@#$%^&*()-+'.indexOf(c) != -1;

const strongPasswordCheckerII = (s) => {
    let n = s.length, hasLower = false, hasUpper = false, hasDigit = false, hasSpecial = false, adjSame = false;
    if (n < 8) return false;
    for (let i = 0; i < n; i++) {
        if (isLowerCase(s[i])) hasLower = true;
        if (isUpperCase(s[i])) hasUpper = true;
        if (isDigit(s[i])) hasDigit = true;
        if (isSpecial(s[i])) hasSpecial = true;
        if (i + 1 < n && s[i] == s[i + 1]) adjSame = true;
    }
    return hasLower && hasUpper && hasDigit && hasSpecial && !adjSame;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/719643327/) More Details (/submissions/detail/719643327/)

Share your acceptance!