

Strong Password Checker

Problem definition

I will create an algorithm that will return the minimum change required to make a string a strong password and if the password is already strong the minimum change will be zero.

Specification of the algorithm

1. It has at least 6 characters and at most 20 characters
2. It must contain at least one lowercase letter, at least one uppercase letter, and at least one digit.
3. It must not contain three repeating characters in a row

The presentation of the algorithm

Firstly, I will verify if the string has lower than 6 characters and there will be more cases:

1. If the string has lower than 3 characters, that means the minimum change required are 6 – the number of the characters from the string, because we cannot have three repeating characters and we can add a lowercase letter, uppercase letter, and one digit even if we have special characters(like # @ \$...) in the string.
2. If the string has 3 characters and if there are 3 repeating characters we have to make 3 insertions and one replace that means it will be 4 changes minimum required. If there are no repeated characters we have only to insert 3 characters.
3. If the string has 4 characters we will verify if there are any repeated characters and if it is true we will need to make 2 insertions and one change. If not I will see if exists at least one lowercase/one uppercase/digit, because if we don't have any of them that means we need to make 2 insertions and replace one character.
4. If the string has 5 characters I will check if there are not the characters that we need and if not it will be one insertion and 2 replace (when all are special characters). If there are all characters that we need we will have one insertion

If there are all characters and there are no repeated characters we will have one insertion. If not no matter that hasn't all characters and we have repeated characters we will make one replace and one add.

Secondly, I will go through all the string and verified if there are repeated characters.

If there are 3/4/5 repeated characters we will make one replace.

Also, I verified if the string has more than 20 characters and find the number of the characters we should delete. We will have more cases:

1. If the number of repeated characters is higher than the number of deleted characters we have other cases :

- we will return the number of the repeated characters because we can delete the characters which are repeating, but still, we will have more deleted characters that we have to replace.

- of course, there can be repeated characters but no the all characters we need and I verified there are any of them and I add to the number of minimum change plus 1/2/3 in function how many characters are missing.

2. If the number of repeated characters is lower than the number of the deleted characters we have other cases:

- we will return the number of the characters we should delete because we will delete the repeated characters.

- of course, we will delete the repeated characters but still, we will have extra characters and that's why we will return this number.

- also if we have to delete some characters and we don't have all the characters that we need to make the password strong we will plus 1/2/3 in the function of how many essential characters are missing.

Also, if the string has at least 6 and at most 20 characters we will verify if we have some repeated characters and there are some cases:

1.if the number of repeated characters is 1, we will see what are the characters there are missing :

- if there are 3 we will return 3, because we can replace one of the repeated characters and replace 2 from the string

-if there are 2 we will return 2, because we will replace one of the repeated characters and one from the string

-if there are 1 we will return 1, because we will replace one of the repeated characters.

-if there are 0 we will return 1, because we have one group of repeated characters in a row

2.if the number of repeated characters is 2, we will see what are the characters there are missing :

-if there are 3 we will return 3, because we will replace 2 repeated characters and one from the string

-if there are 2 we will return 2, because we will replace both groups of 3 characters in a row.

-if there are 1 we will return 2, because we have to replace one character in one group of three repeated characters in a row and delete/replace one character from the second group of repeated characters.

-if there are 0 we will return 2, because we have to replace/delete one character in one group of three repeated characters in a row and delete/replace one character from the second group of repeated characters.

3.if the number of characters is more or equal with 3 we will return the number of the group with repeated characters because no matter what how characters are missing we can replace them in the repeated group of characters.

Finally, if there are not repeated elements or we have not to delete other characters we will check if there are any characters that we need and if not we will return the number of the missing characters.

Program Testing

There I will test some special passwords to see if the program will return the correct value.

Password : Ovidiu1122 return 0;

Password : aaB1 return 2;

Password : aaaB1 return 2;

Password : aaabbbcccdD1dddDDD111aaaaa return 11;

Password : #@\$12 return 2;

Password : \$\$\$1 return 3;

Password : ## return 4;

Password : aaaaa return 2;

Password : Ovidiuuuu1 return 1;

Password : uuuxddaA return 1;

Result

I create an algorithm that will verify that each password will respect all points of the specification algorithm.