Problem #2299: Strong Password Checker II. (Easy)

<https://leetcode.com/problems/strong-password-checker-ii/description/>

My Solution:

1. If length of password is less than 8, then return False
2. Iterate through password from character at index 1 to the end of the string.

Compare each character to the previous character. If they are the same, return False.

1. Initialize lowercase, uppercase, has Digitm hasSpecialChar to False.
2. Let specialChars be the set of special characters "!@#$%^&\*()-+".
3. Now iterate through password character by character. If the present character ch is lowercase then lowercase is True, else if it is uppercase then uppercase is True,else if it is a digit, then hasDigit is True, else if is a special character then hasSpecialChar is True.

Check whether lowercase, uppercase hasDigit and hasSpecialChar are all True. If so, return True.

1. Return False in the end.

class Solution:

def strongPasswordCheckerII(self, password: str) -> bool:

if len(password) < 8:

return False

for i in range(1, len(password)):

if password[i] == password[i - 1]:

return False

lowercase = False

uppercase = False

hasDigit = False

hasSpecialChar = False

specialChars = set("!@#$%^&\*()-+")

for ch in password:

if ch.islower():

lowercase = True

elif ch.isupper():

uppercase = True

elif ch.isdigit():

hasDigit = True

elif ch in specialChars:

hasSpecialChar = True

if lowercase and uppercase and hasDigit and hasSpecialChar:

return True

return False