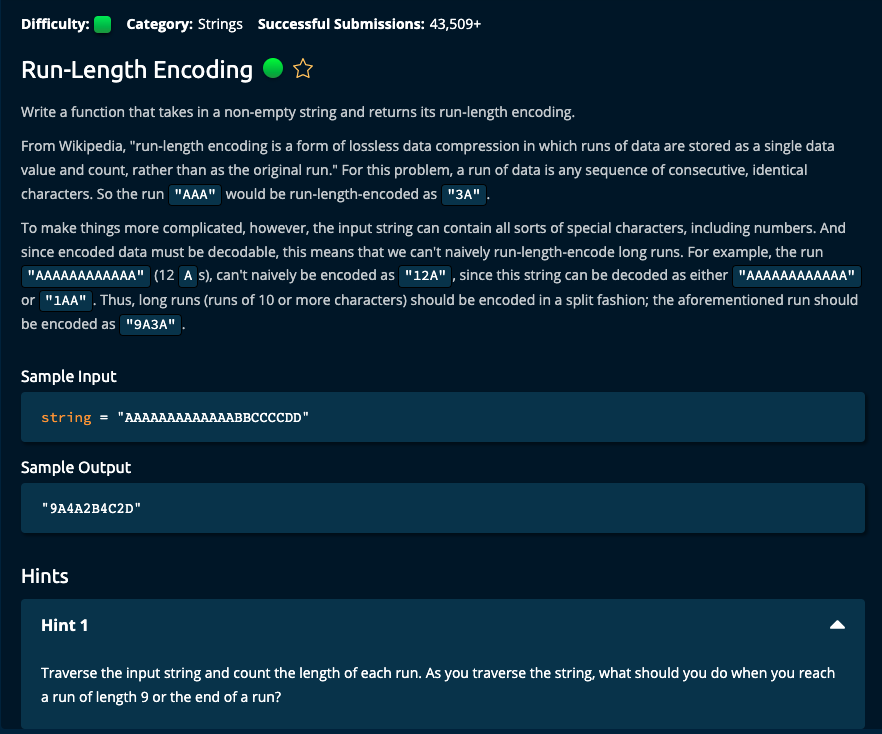
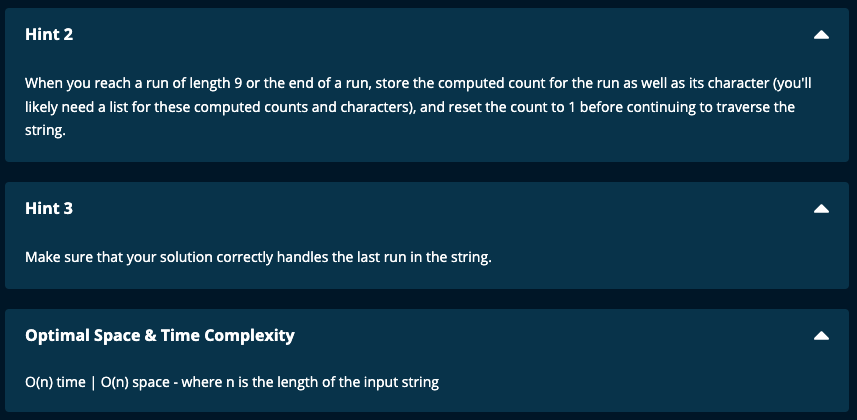
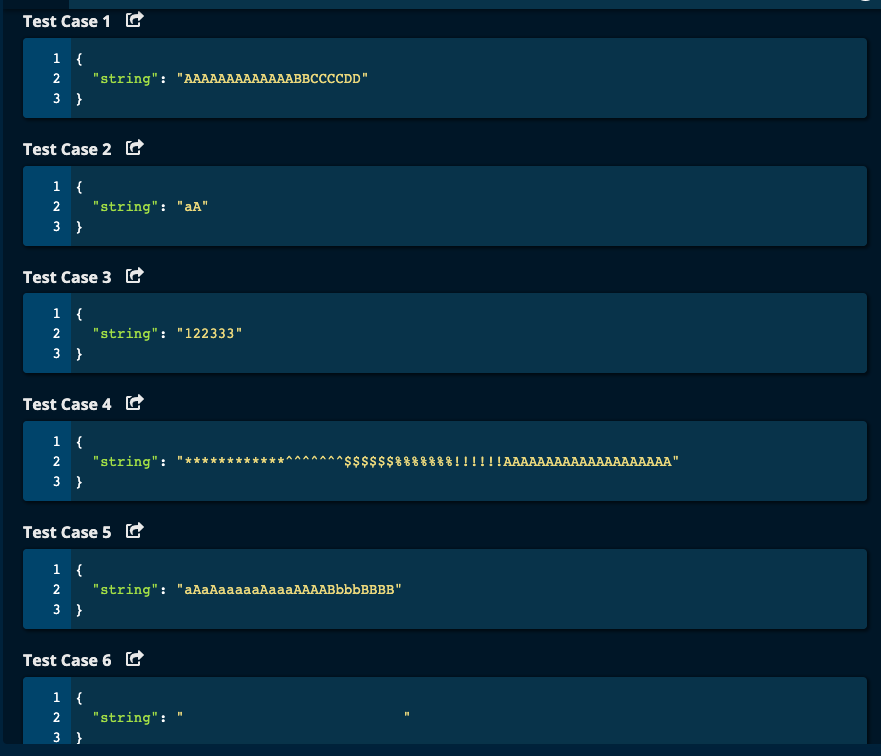
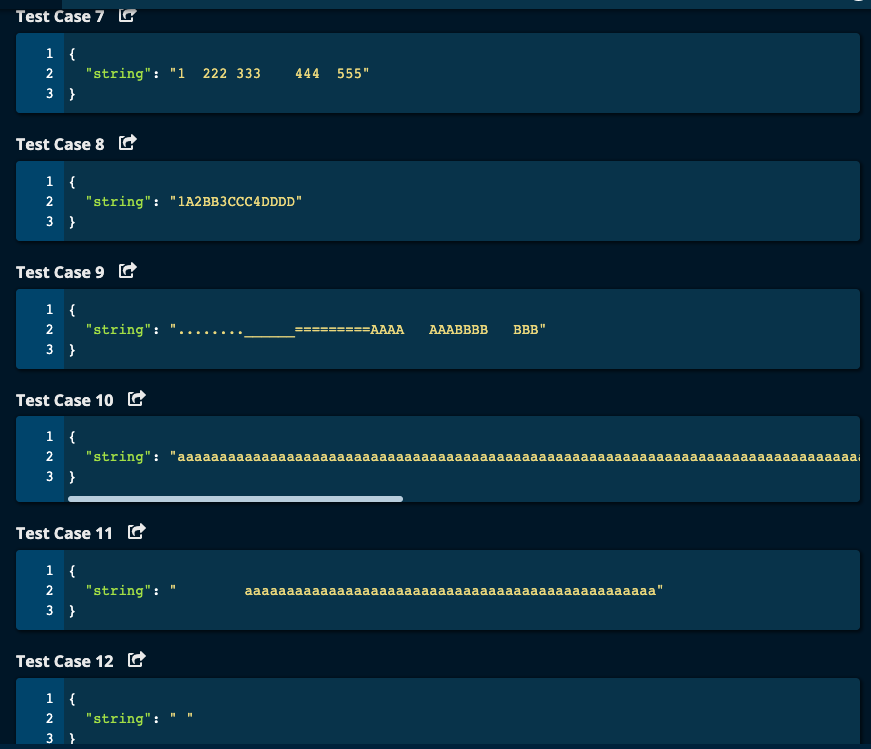
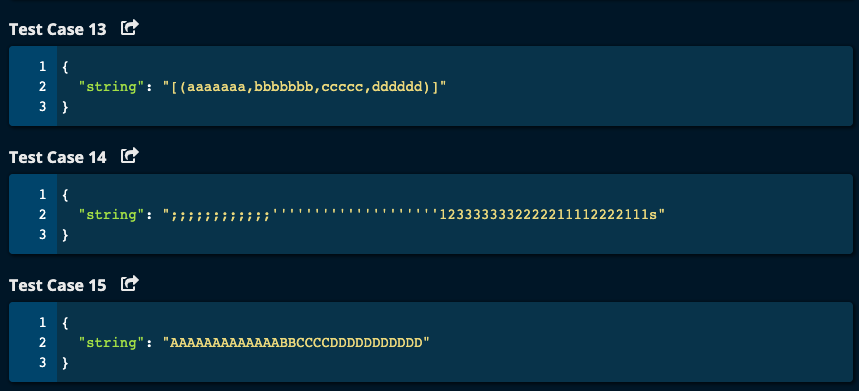
Run Length Encoding (Easy)











My Solution:

# My Solution -- O(n) Time | O(n) Space

def runLengthEncoding(string):

result = [] # to store run encoding

runLength = 1

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

prevChar = string[i - 1]

currChar = string[i]

if prevChar != currChar or runLength == 9:

result.append(str(runLength))

result.append(prevChar)

runLength = 1

else:

runLength += 1

#Handling end of the string case

result.append(str(runLength))

result.append(string[-1]) #last character, i.e. string[len(string) - 1]

return "".join(result)

-----------------------------------------

JJ Notes:

1. Initialize an empty list called result to hold the encoded string.
2. Initialize runLength to 1.
3. Iterate through string array from index 1. The previous character, prevChar is the string value at the previous index and currChar is the string value at the current index.

If the prevChar and currChar are not equal or if runLength is 9, then convert the runLength to a char and append it to result. Also append the previous character to result.

If the prevChar and currChar are the same and if runLength is less than 9 (i.e. the else condition), just increment the runLength.

1. Handle the end of string case by appending the runLength converted to a string and appending the last character of the string.
2. Convert the result list to a string and return it.

Algoexpert Solution:

JJ Notes:

1. Almost the same as my solution. In the for loop, if condition for previous character not the same as current character or currentRunLength is equal to 9, then after appending run length and the previous character to the result list, the currentRunLength is reset to 0.
2. Then currentRunLength is incremented by 1 without any conditions. So we don’t need an else statement.
3. The rest is all the same logic.

