Problem # 784: Letter case Permutation (Medium)

https://leetcode.com/problems/letter-case-permutation/

My Solution:

Runtime beats 23.95%

class Solution:

def letterCasePermutation(self, S: str) -> List[str]:

n = len(S)

if n == 0:

return []

count = 0 # count the number of letters

for i in range(n):

if S[i].isalpha():

count += 1

num\_combos = 2 \*\* count # number of combinations of upper and lower case letters

result = []

for combo in range(num\_combos):

combo = ""

result.append(combo)

count\_letter = 0

for i in range(n):

if S[i].isalpha():

count\_letter += 1

num\_groups = 2 \*\* count\_letter

group\_size = num\_combos // num\_groups

for j in range(num\_combos):

group\_num = j // group\_size

if group\_num % 2 == 0:

result[j] = result[j] + S[i].lower()

else:

result[j] = result[j] + S[i].upper()

else:

for j in range(num\_combos):

result[j] = result[j] + S[i]

return result