**Problem #1370: Increasing Decreasing String. (Easy)**

<https://leetcode.com/problems/increasing-decreasing-string/description/>

My Solution:

1. If the length of s is 0, then return s.
2. Let counter\_s be the frequency dictionary of s.
3. Let keys be the keys in the dictionary counter\_s.
4. Let inc\_sorted\_keys be the keys sorted in increasing order in a list keys
5. Let dec\_sorted\_keys be the keys sorted in decreasing order in the list keys.
6. Initialize res to be an empty list to hold the results.
7. Let num\_keys be the length of counter\_s.
8. While num\_keys is greater than 0, traverse through the keys in inc\_sorted\_keys.

If the key is counter\_s.keys(), then append the key to res. Decrement the value of the key in counter\_s. If the value of the key in counter\_s reaches 0, then delete the key-value pair in counter\_s.

1. Traverse through the keys in dec\_sorted\_keys.

If the key is counter\_s.keys(), then append the key to res. Decrement the value of the key in counter\_s. If the value of the key in counter\_s reaches 0, then delete the key-value pair in counter\_s.

1. Join res without spaces and return the string.

====================================

from collections import Counter

class Solution:

def sortString(self, s: str) -> str:

if len(s) == 0:

return s

counter\_s = Counter(s)

#print("s = ", s, "counter\_s = ", counter\_s)

keys = counter\_s.keys()

inc\_sorted\_keys = sorted(keys)

dec\_sorted\_keys = inc\_sorted\_keys[::-1]

res = []

num\_keys = len(counter\_s)

while num\_keys > 0:

for key in inc\_sorted\_keys:

if key in counter\_s.keys():

res.append(key)

counter\_s[key] -= 1

if counter\_s[key] == 0:

del counter\_s[key]

num\_keys -= 1

for key in dec\_sorted\_keys:

if key in counter\_s.keys():

res.append(key)

counter\_s[key] -= 1

if counter\_s[key] == 0:

del counter\_s[key]

num\_keys -= 1

return "".join(res)