**Justin\_String\_1081. Smallest Subsequence of Distinct Characters**

**Concept:**

先算出每個字母出現幾次

用 stack 法

一一搜尋，如果沒出現過，則從 stack 由上至下比較

如果比較 stack 小，而且 stack 的字母後面還有，則可以替換

如果出現過，則直接看替換會不會變小

**Code:**

class Solution:

def smallestSubsequence(self, text: str) -> str:

result = ""

dic = {}

dic\_count = {}

for i in range(len(text)):

if str(text[i]) not in dic\_count:

dic\_count[str(text[i])] = 1

else:

dic\_count[str(text[i])] += 1

for j in range(len(text)):

if str(text[j]) not in result:

delete\_num = 0

for k in range(1, len(result)+1):

if str(text[j]) < str(result[-k]) and dic\_count[str(result[-k])] > 0:

delete\_num += 1

else:

break

if delete\_num > 0:

result = result[:-delete\_num]

result += str(text[j])

else:

if result > result.replace(str(text[j]), "") + str(text[j]):

result = result.replace(str(text[j]), "") + str(text[j])

dic\_count[str(text[j])] -= 1

return result