Week 13

Topic: Stack

1209. Remove All Adjacent Duplicates in String II

- Difficulty: Medium
- Problem URL: https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string-ii/
- Description:
  給定一個字串 s 和一個整數 k，每次移除 k 個相鄰且相同的字母，直到無法再移除為。最後把剩下的字串重新接在一起。

Example1:

Input: s = "abcd", k = 2
Output: "abcd"
Explanation: There's nothing to delete.

Example2:

Input: s = "deeedbbcccbdaa", k = 3
Output: "aa"
Explanation:
First delete "eee" and "ccc", get "ddbbbdaa"
Then delete "bbb", get "dddaa"
Finally delete "ddd", get "aa"

Example3:

Input: s = "pbbcggttciiippooaais", k = 2
Output: "ps"

詳細說明與約束條件請參考 Leetcode 網站。

---

316. remove-duplicate-letters

- Difficulty: Medium
- Problem URL: https://leetcode.com/problems/remove-duplicate-letters/
- Description:
  給定一個字串 s，請移除重複字母，以達到每個字母只出現一次，並在所有符合條件的組合中，回傳字典序 (Lexicographical Order) 最小的那個字串作為答案，同時不能不能打亂原來字母的順序。

Example1:

> Input: s = "bcabc"
> Output: "abc"

Example2:

> Input: s = "cbacdcbc"
> Output: "acdb"

詳細說明與約束條件請參考 Leetcode 網站。

---

895. maximum-frequency-stack

- Difficulty: Hard
- Problem URL: https://leetcode.com/problems/maximum-frequency-stack/
- Description:
  設計一個特殊的 Stack，除了 push(val) 以外，pop() 時要根據「元素出現的頻率」來決定要移除哪個元素，若有多個頻率一樣，則回傳「最接近 stack 頂端的那個元素」。

Example1:

Input:

["FreqStack", "push", "push", "push", "push", "push", "push", "pop", "pop", "pop", "pop"]

[[ ], [5], [7], [5], [7], [4], [5], [ ], [ ], [ ], [ ]]

Output:

[null, null, null, null, null, null, null, 5, 7, 5, 4]

Explanation:

FreqStack freqStack = new FreqStack();	
freqStack.push(5);	// The stack is [5]
freqStack.push(7);	// The stack is [5,7]
freqStack.push(5);	// The stack is [5,7,5]
freqStack.push(7);	// The stack is [5,7,5,7]
freqStack.push(4);	// The stack is [5,7,5,7,4]
freqStack.push(5);	// The stack is [5,7,5,7,4,5]
freqStack.pop();	// return 5, as 5 is the most frequent. The stack becomes [5,7,5,7,4].
freqStack.pop();	// return 7, as 5 and 7 is the most frequent, but 7 is closest to the top. The stack becomes [5,7,5,4].
freqStack.pop();	// return 5, as 5 is the most frequent. The stack becomes [5,7,4].
freqStack.pop();	// return 4, as 4, 5 and 7 is the most frequent, but 4 is closest to the top. The stack becomes [5,7].

詳細說明與約束條件請參考 Leetcode 網站。
