

Week 1

主題: 資料結構

146. LRU Cache

- Difficulty: Medium
- 題目: [Leetcode 146 - LRU Cache](#)
- 題目描述:
LRU Cache (Least Recently Used)是一種快取策略，當容量達到上限時，會移除最久未使用的數據。我們需要實作 `LRUCache` 類別，提供 `get(key)` 取得值(若不存在回傳 `-1`)，以及 `put(key, value)` 新增或更新數據(若超過容量則移除 LRU 鍵)。所有操作的平均時間複雜度必須為 $O(1)$ 。

Example:

Input:

- ["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"]
- [[2], [1, 1], [2, 2], [1], [3, 3], [2], [4, 4], [1], [3], [4]]

Output: [null, null, null, 1, null, -1, null, -1, 3, 4]

Explanation:

```
LRUCache lruCache = new LRUCache(2);
lruCache.put(1, 1); // cache is {1=1}
lruCache.put(2, 2); // cache is {1=1, 2=2}
lruCache.get(1);    // return 1
lruCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}
lruCache.get(2);    // returns -1 (not found)
lruCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}
lruCache.get(1);    // return -1 (not found)
lruCache.get(3);    // return 3
lruCache.get(4);    // return 4
```

詳細說明與約束條件請參考 [Leetcode](#) 網站。

460. LFU Cache

- Difficulty: Hard
- 題目: [Leetcode 460 - LFU Cache](#)
- 題目描述:

LFU Cache(Least Frequently Used)是一種快取策略，當容量達到上限時，會移除使用頻率最低的數據。我們需要實作 LFU Cache 類別，提供 `get(key)` 取得值(若不存在回傳 -1)，以及 `put(key, value)` 新增或更新數據(若超過容量則移除使用頻率最低的鍵)。若有多個鍵頻率相同，則淘汰最久未使用的鍵。每個鍵都有一個使用計數器，在 `get` 或 `put` 操作時增加。所有操作的平均時間複雜度必須為 $O(1)$ 。

Example

Input:

- ["LFUCache", "put", "put", "get", "put", "get", "get", "put", "get", "get", "get"]
- [[2], [1, 1], [2, 2], [1], [3, 3], [2], [3], [4, 4], [1], [3], [4]]

Output: [null, null, null, 1, null, -1, 3, null, -1, 3, 4]

Explanation:

詳細說明與約束條件請參考 [Leetcode](#) 網站。

239. Sliding Window Maximum

- Difficulty: Hard
- 題目: [Leetcode 239 - Sliding Window Maximum](#)
- 題目描述:

給你一個整數陣列 `nums`，有一個尺寸為 `k` 的視窗從陣列的最左側開始滑動，目的是找出每次視窗移動時的最大值。每次視窗只能向右移動一位置，你需要每次將目前視窗內最大的值進行輸出。

Example:

Input: `nums = [1,3,-1,-3,5,3,6,7]`, `k = 3`

Output: `[3,3,5,5,6,7]`

Explanation:

Window Position	Max
[1 3 -1] -3 5 3 6 7	3
1 [3 -1 -3] 5 3 6 7	3
1 3 [-1 -3 5] 3 6 7	5
1 3 -1 [-3 5 3] 6 7	5
1 3 -1 -3 [5 3 6] 7	6
1 3 -1 -3 5 [3 6 7]	7

詳細說明與約束條件請參考 [Leetcode 網站](#)。