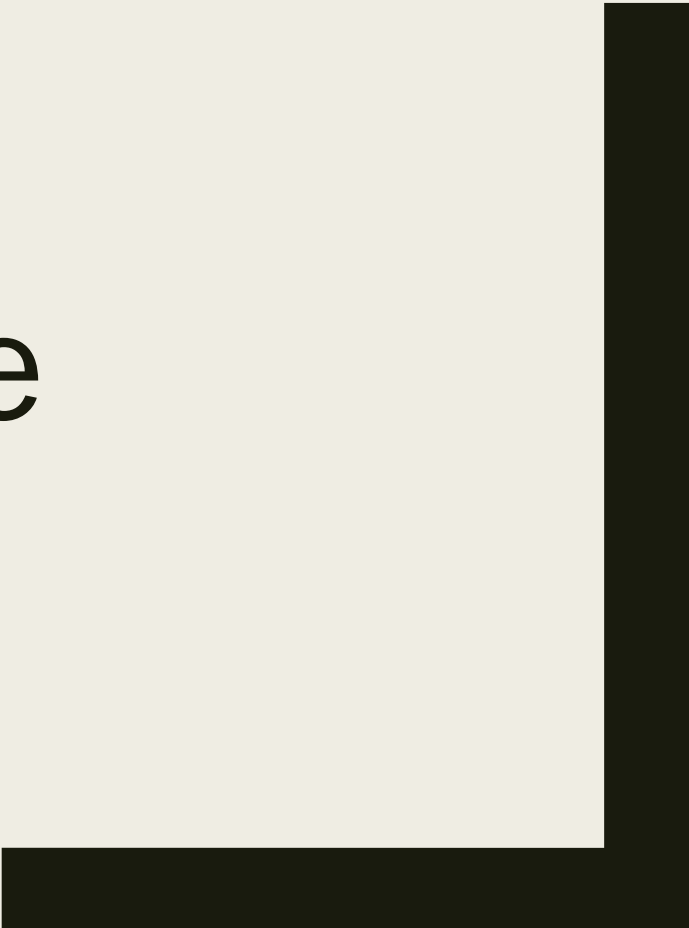


補充

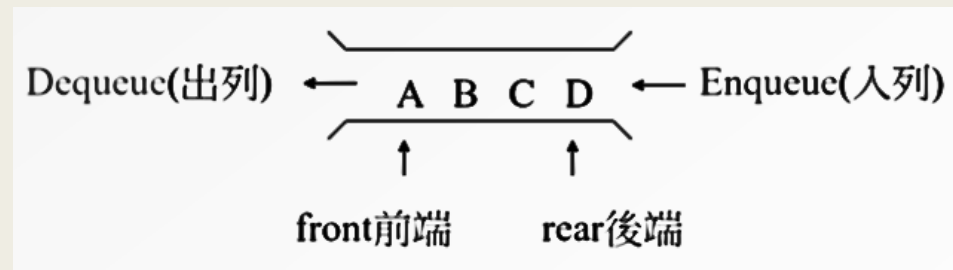
Priority Queue

(優先佇列)

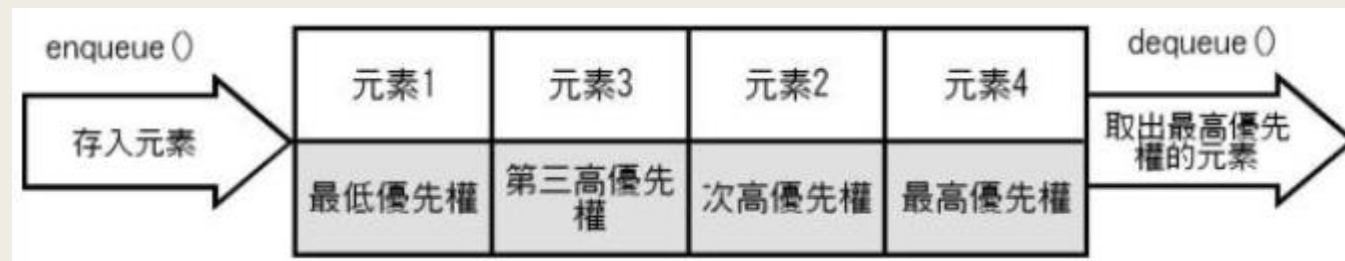


Priority Queue

- 佇列(Queue)：先進先出(FIFO)，沒有優先權。應用：排隊買票



- 優先佇列：儲存的每個元素會指定一個優先權(Priority)，取出順序並不是依據先後順序，而是最高優先權者先取出來。應用：醫院的急診室，最嚴重的病患先處理；HW; ...

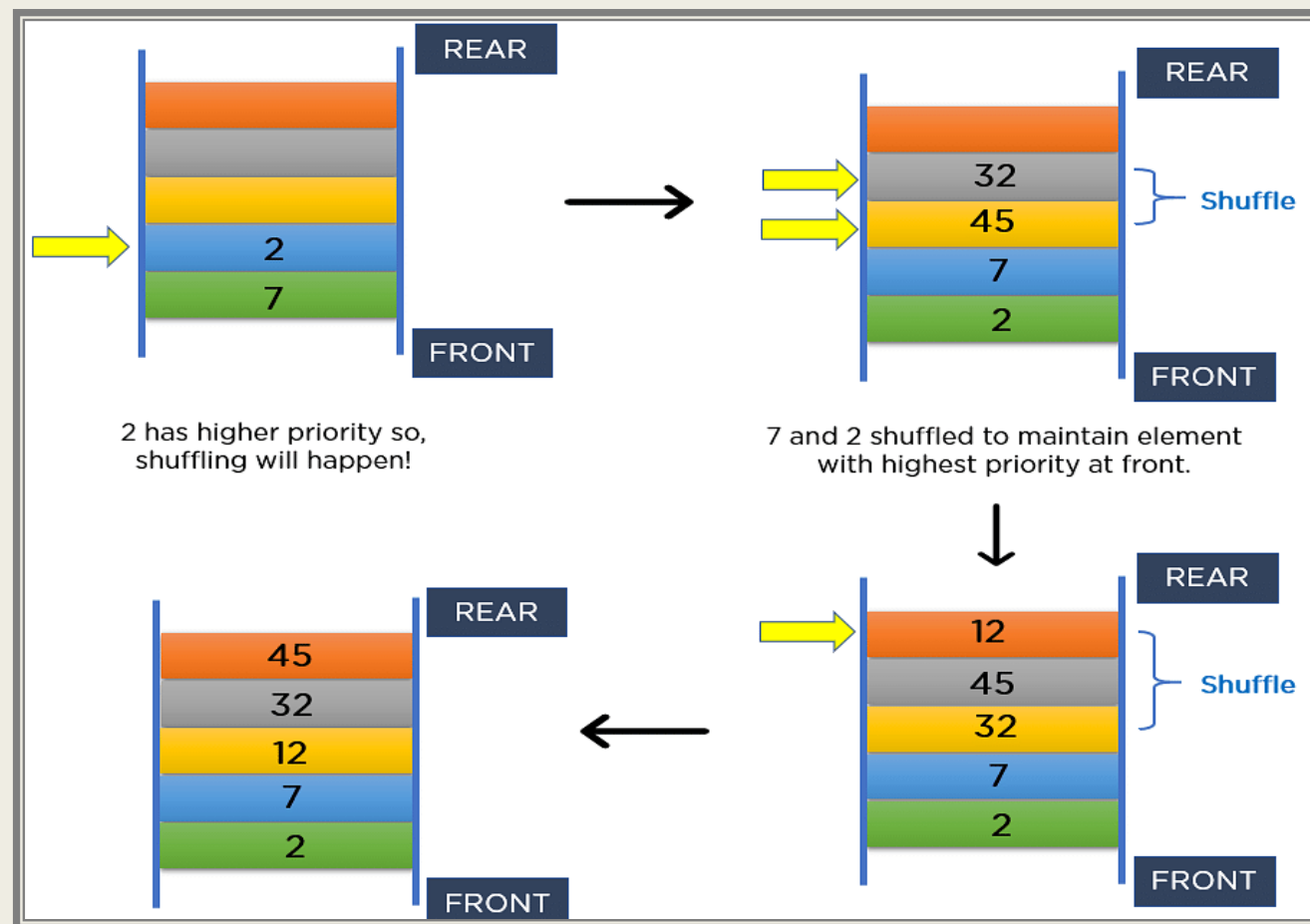


(Max-Priority Queue)

PS: 若取出時考慮優先權最小的，則稱為Min-Priority Queue

Priority Queue

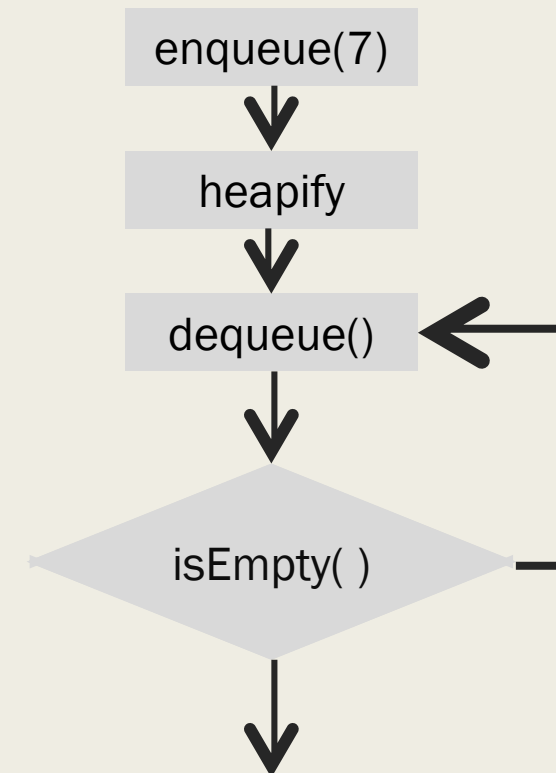
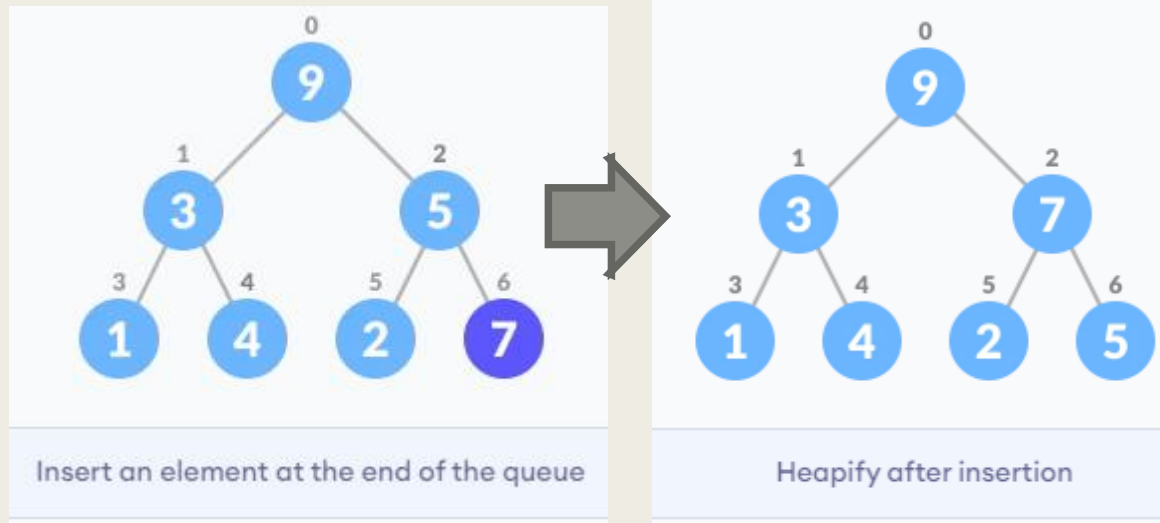
- Max-Priority Queue有三個操作
 - Insert: 將資料加入queue中
 $O(\log n)$
 - IncreaseKey: 依某元素的優先權提高其重要性
 $O(\log n)$
 - ExtractMax: 取出最重要的元素
 $O(1)$
並從queue中移除
 $O(\log n)$



Priority Queue

函數	說明
int isPriorityQueueEmpty()	檢查優先佇列是否是空的，如果是，傳回 1，否則為 0
void enqueue(int d)	將參數的元素存入優先佇列，也就是將元素插入堆積
int dequeue()	從優先佇列取出元素，取出的是目前佇列中最高優先權的元素，也就是堆積的根節點

- 實作：使用最大堆積樹 (Max Heap Tree)



Priority Queue

- 實作：C++已有優先佇列標準函式庫 (priority_queue)

```
#include <queue>
#include <iostream>
using namespace std;

int main(void) {
    priority_queue<int> q;
    q.push(5);
    q.push(9);
    q.push(14);
    q.push(3);
    q.push(27);
    q.push(1);
    cout << "佇列元素各數" << q.size() << endl;
    while (!q.empty()) {
        cout << q.top() << " ";
        q.pop();
    }
    cout << endl;
    return 0;
}
```

C:\Users\user\Desktop\Heap3.exe

佇列元素各數6
27 14 9 5 3 1

Process exited after 0.04126 seconds with return value 0
請按任意鍵繼續 . . .