

第十五週

王子銓, 陳毅軒, 吳尚龍

電機通訊程式設計

May 27, 2024

Outline

1 Queue

2 Queue 實作

Queue

queue (Queue) 是一種線性資料結構，其特點是遵循「先進先出」(First In, First Out, FIFO) 的原則。也就是說，最早加入的元素最先被移除。

queue 廣泛應用於許多計算機科學領域，包括：

- 任務排程：在作業系統中管理進程。
- 資料緩衝：在網路資料傳輸中作為緩衝區。
- 寬度優先搜索（BFS）：圖算法中的一種遍歷方法。

Queue 實作

queue 的主要操作：

- Enqueue：將元素放入 queue 的尾端。
- Dequeue：將 queue 頭端的元素移除。
- Front：查看 queue 頭端的元素，但不移除它。
- IsEmpty：檢查 queue 是否為空。

上上週教過 Linked list，Queue 其實就是一個在 tail 進行 Enqueue，在 head 進行 Dequeue 的 Linked list，相信大家會刻 Linked list 就會刻 Queue。

Array 實作

```
1 #define CAPACITY 100 // 定義queue的最大容量
2 typedef struct {
3     int front, rear, size;
4     int data[CAPACITY];
5 } QueueArray;
```

陣列實作會比 Linked list 簡單很多，但缺點是必須事先定義 queue 的最大容量。因為大小有限，最好多實作一個檢查已滿的函式。

```
1 // 檢查queue是否已滿
2 int isFull(QueueArray* queue) {
3     return queue->size == CAPACITY;
4 }
```


比較優缺點

使用陣列實作 queue

- 陣列實作 queue 的優點是簡單且訪問速度快，但缺點是必須事先定義 queue 的最大容量。以及當 dequeue 後，浪費記憶體的問題。

使用鏈結串列實作 queue

- 鏈結串列實作 queue 的優點是沒有容量限制，缺點是相對陣列來說，每個節點需要額外的記憶體來存儲指標。

queue 操作的時間複雜度

queue 的各種操作（Enqueue、Dequeue、Front、IsEmpty）時間複雜度均為 $O(1)$ 。這是因為這些操作只涉及對 queue 頭尾元素的操作，無論 queue 大小，都能在常數時間內完成。

教學意見調查表

★ 請同學填寫**教學意見調查表**，若填寫率超過 90%，總成績 +2 分 (14 次作業 *2+ 教學意見調查表 =30% 作業成績)