

第十四週

王子銓, 陳毅軒, 吳尚龍

電機通訊程式設計

May 20, 2024

Outline

1 Stack

2 Stack 實作

Stack

堆疊（Stack）是一種線性資料結構，其特點是遵循「後進先出」（Last In, First Out，LIFO）的原則。也就是說，最新加入的元素最先被移除。

堆疊廣泛應用於許多計算機科學領域，包括：

- 函數呼叫堆疊：在程序執行期間，追蹤函數呼叫。
- 表達式求值：例如，中序（infix）轉後序（postfix）表示法和其求值。
- 回溯演算法：例如，深度優先搜索（DFS）。

Stack 實作

堆疊的主要操作：

- Push：將元素放入堆疊的頂端。
- Pop：將堆疊頂端的元素移除。
- Top：查看堆疊頂端的元素，但不移除它。
- IsEmpty：檢查堆疊是否為空。

上週教過 Linked list，Stack 其實就是只對 head 做操作的 Linked list，相信大家會刻 Linked list 就會刻 Stack。

Array 實作

```
1 #define CAPACITY 100 // 定義堆疊的最大容量
2 typedef struct {
3     int top;
4     int data[CAPACITY];
5 } StackArray;
```

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

```
1 // 檢查堆疊是否已滿
2 int isFull(StackArray* stack) {
3     return stack->top == CAPACITY - 1;
4 }
```


比較優缺點

使用陣列實作堆疊

- 陣列實作堆疊的優點是簡單且訪問速度快，但缺點是必須事先定義堆疊的最大容量。

使用鏈結串列實作堆疊

- 鏈結串列實作堆疊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。

堆疊操作的時間複雜度

堆疊的各種操作（Push、Pop、Top、IsEmpty）時間複雜度均為 $O(1)$ 。這是因為這些操作只涉及對堆疊頂端元素的操作，無論堆疊大小，都能在常數時間內完成。