**# Data Structure**

\* Question

- 課本有的但未納入老師ppt的小節會考嗎？

eg.Ch4 4.11 Generalized List

考試題目需要用到的概念都是課堂教過的，不會刻意出很冷門題目

**## Ch2 Array**

### KMP Algorithm

\* Question

- PPT P.33 • O(columns × terms) vs. \*\*O(columns × rows)\*\*

- PPT P.61 O(n) 指的是? prefixTable[]?

O(n)為線性時間，如for(i=1 to n)逐一操作的時間

Prefix table為kmp精髓，建議上網搜尋相關教學

**## Ch3 Stacks & Queues**

\* Question

- stack或queues的 pop()、push()是c++本身就有的函示，還是programmer要自己寫？

本身就有

- 想請問用stack來表示postfix相較其他方法有什麼優點？

- 有pre-order以及in-order就得出unique的binary tree

=> 那pre-order & post-order可以嗎？

=> 那in-order & post-order可以嗎？

必須有in-order才可以決定唯一

**## Ch4 Listed Node**

\* Questions

\* PPT 9 - 11 3個 implement method\*\*P.11\*\*

\* PPT 15 想請問 Nested classes有什麼樣的好處？(因為在閱讀時，發現比較難懂，所以不明白為什麼有這種寫法。)

可以參考 https://openhome.cc/Gossip/CppGossip/NestedClasses.html 的解說

\* p.12-15 composite classes V.S. Nested classes

Composite class 參考--> https://iter01.com/475772.html

\* Composite classes : 把Node, List operations 分開寫在兩個不同classes

\* \*\*(???)Nested classes\*\* Node, List operations 寫在同一個classes。private 裡面看不懂

\* p.16 \*x = \* y ---> 內容之間的assign也可以，所以\*y 存 b，\*x的內容也會變b

\* p.21 - 23 ---->網路上查詢template相關知識

\* p.29 code

第一個點只有自己所以全部指自己

其餘情況:

Circular由front插入，原本末端點last的last->link為front所以要把

newNode->link指向last->link;

而原本last->link則指向新加入的newNode;

\* p.32 delete delNode; c++語法

\* PPT 35 想請問最後一行的 LIST是什麼？是c++中的一種資料型態嗎？是

\* P.35 code

\* PPT 39 想請問關於第一點，為什麼circular list會比較有效率？(不管circular，還是linear在插入或是刪除node時，都要重新連接前後的nodes呀)

這句的重點不是在增刪而是頻繁，circular可以從任何點開始並能抵達circular上任一節點，但singly則一定要回溯到最前端，在一些作業系統的process中效率就會有差

\* PPT 40 不太懂這張投影片在述說什麼？因此code也看不懂

\* p.42 x 後面的怎麼辦？

\* p.48 equlvalence relations 最後是怎麼一組一組推出來的

檢查反身、對稱、遞移三特性，皆符合即為等價關係（詳見離散數學）

**## Ch5 Trees**

\* Questions

\* PPT 9 看不太懂這張的意思是什麼，在猜想是不是與 node array表示法有關？

每格n點的k-ary tree共有nk個child field，除了leaf(子全空)每個點至少一個非空子節點所以只有k-1個空，所以(n-1)(k-1) + k =nk-n+1=n(k-1)+1

\* PPT 14 想請問最後一行的 the order of the subtree is irrelevant是什麼意思？

一般tree的子樹間沒有順序，也就是沒有左右子樹分別

\* P.17 lemma 5.3(P.19)

這是公式請背起來，沒有為什麼

\* PPT 68 不懂三個 time complexity為什麼是這樣的數值？

Heap delete max/min本身為O(1),重新調整heap需要層層檢查，因此複雜度為同樹高 ,N為節點數

Heap刪除非root便需要重建整棵樹，因此為O(N)

Heap無法用分段搜尋所以要全部n個都看過O(N)