單元4: 以遞迴解題

Defining Language

A language — A set of strings of symbols
(語言) — Ex: English、C++

A grammar — The rules for forming the string in a language
(文法語法) — Ex: English grammar, C++ syntax rule

Algebraic Expressions

Intix expressions 中序
Ex: $a + b$ $((a + b) * c)/d$ // 順序問題

Prefix expressions 前序
Ex: $+ a b$ $/ * + a b c d$ // 前、後序則 無

Posttix expressions 後序   順序問題.
Ex: $a b +$ $ab + c * d/$

Fully Parenthesized Expressions

Fully parenthesized intix expressions
— Do not require precedence rules or rules for association

— But are inconvenient for programmers

Remark: 以中序來寫會有先後順序問題,需以括弧解決。

優 較有彈性的擴充 → 有效運用資源

缺 沒處理好資料 → 容易資料不見

Array has a fixed size.
　　陣列 (需移動資料)
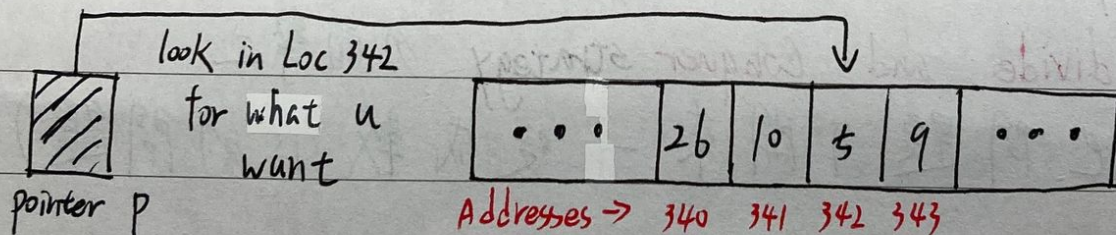
Linked List is able to grow in size as needed.
　　鏈結串列 (不需移動資料)

Pointers

　　Declaration of an integer pointer variable P

　　int *p ;

pointer P

look in Loc 342
for what u
want

... | 26 | 10 | 5 | 9 | ...

Addresses → 340　341　342　343

The new operator

　　P = new int ;　　// 申請一棟新房子

　　delete P ;　　// 歸還房子

　　P = NULL ;　　// 遺忘門牌

單元 2 ：抽象化

物件導向概念

Example :   classes of objects
Attributes : data members   (資料
Behaviors : methods / member functions   (程序/動作

Three characteristics

Encapsulation 封裝

- Objects   combine data and operations
- Hides inner details

Inheritance   繼承

- Classes can inherit properties from other classes
- Existing classes can be reused

Polymorphism   多型

- Objects can deter
- execution time

單元 1：遞迴

以相同的方法解決問題

Recursive Functions

Factorial

Greatest Common Divisor

Search in Array

Fibonacci series

Combinatorial numbers

Towers of Hanoi

A binary search is recursive

Use a divide and Conquer strategy 分而擊之

（試著把問題分開來 → 變成較小的問題）

Remark：

1. 把問題拆解成更小的問題

2. 確認 Base Case 有效且可執行

3. 所有 case 最終到 Base case