

資料結構 1-4 章筆記 資訊二乙 林玟妤 11027264

My Notes
Important Concepts worth keeping

遞迴

Today: , ,

- * 將大問題拆解成小問題
- * 遞迴可做的 ex: Factorial 階乘, Great Common divisor 最大公因數, Search in Array 搜尋, Fibonacci series 費氏數列, Combinatorial numbers 組合數, Towers of Hanoi 河內塔
- * 倒著印出

```
void writeBackward (string s, int size) {
    if (size > 0) {
        cout << s.substr (size-1, 1);
        writeBackward (s, size-1);
    }
}
```

$s = \text{'cat'}$
 $size = 3$
 $s = \text{'ca'}$
 $size = 2$
 $s = \text{'c'}$
 $size = 1$

先印出最古老的

We are convinced that love is the principal guiding force in education. We, teachers and students alike, pursue mutual growth through instruction by both words and deeds, in a spirit of love and respect for one another.

4 (CYCU Education Philosophy)

My Questions
Problems & Difficulties needing exploration

遞迴

* 寫一個遞迴函式，讓 a 加到 b ($a > b$)

```
int sum (int a, int b) {
    if (a > b)
        return sum (a-1, b) + a;
    else return b; // a == b
}
```

$100, 99$
 $99, 90$
 $98, 90$
 $90, 90$

* 遞迴 4 步驟

- 遞迴定義 $sum(a+n, b) = sum(a+n, b+1) + b$
- 問題簡化 $b \rightarrow b+1$
- 終止條件 $b, b+1, \dots, a \Rightarrow b = a$
- 保證中止 natural numbers $a > b$

My Opinions
Thoughts, inspirations, and suggestions

我們相信「愛」是教育的主導力量，願以身教言教的方式，互愛互敬的態度，師生共同追求成長。

5 (中興大學教育理念)

My Notes
Important Concepts worth keeping

Greatest Common Divisor

Today: , ,

- * 求最大公因數

```
int gcd1 (int x, int y) {
    if (y == 0) return x;
    else if (y > x) return gcd1 (x, y % x);
    else return gcd2 (y, x % y);
}

int gcd2 (int x, int y) {
    if (! (x % y)) return y;
    else return gcd2 (y, x % y);
}
```

$x = 9, y = 6$
 $gcd1(6, 3) = ?$
 $x = 6, y = 3$
 $gcd1(3, 0) = ?$
 $x = 3, y = 0, \text{return } 3$

$x = 9, y = 6$
 $gcd2(6, 3) = ?$
 $x = 6, y = 3$
 $\text{return } 3$

當 $x < y$ $gcd1$ 效率 = $gcd2$ 效率
 $x > y$ $gcd1$ 效率 < $gcd2$ 效率

Faith, Hope and Love!
Faith lays the foundation,
Hope brings sunshine,
Love changes everything.

6 Sunday

My Questions
Problems & Difficulties needing exploration

Binary Search with an Array

```
int binarySearch (const int anArray[], int first, last, value) {
    int index;
    if (first > last) index = -1; // 方法是否到終止
    else {
        int mid = (first + last) / 2;
        if (value == anArray[mid]) index = mid;
        else if (value < anArray[mid]) // 求解：左半邊
            index = binarySearch (anArray, first, mid-1, value);
        else // 求解：右半邊
            index = binarySearch (anArray, mid+1, last, value);
    }
    return index;
}
```

My Opinions
Thoughts, inspirations, and suggestions

陽光達人傳真愛
信實可築打鐵基礎
樂觀希望散發正能量
真心關愛可以改變一切

密碼 cypher

7 三葉

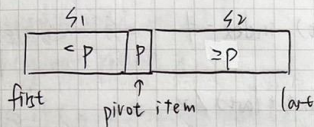
My Notes

Important Concepts worth keeping

找 k^{th} smallest Item in an Array

Today:

* 找 pivot item (樞紐點) in the array



* 只找其中一

k^{Small} (k , anArray, first, last)

= k^{Small} (k , anArray, first, pivotIndex - 1) // if $k < \text{pivotIndex} - \text{first} + 1$ 若落在半區

= P // if $k = \text{pivotIndex} - \text{first} + 1$

= k^{Small} ($k - (\text{pivotIndex} - \text{first} + 1)$, anArray, pivotIndex + 1, last) // if $k > \text{pivotIndex} - \text{first} + 1$

Ex: $k=4$, pivot item: anArray[first]

(39) 7 25 39 19 48 16 12

= (39) 7 25 19 16 48 39

= (12) 7 25 19 16 30 48 39

= (12) 7 25 19 16 19 25

ONCE, all the villagers decided to pray for rain. On the day of prayer all the people gathered, but only one boy came with an umbrella.

That's FAITH!

-Anonymous

My Questions

Problems & Difficulties needing exploration

Reversing an Array 反轉元素次序

Algorithm ReverseArray (anArray, low, high) {

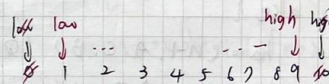
if ($\text{low} < \text{high}$) then (虛擬碼)

Swap anArray[low] and anArray[high]

ReverseArray (anArray, low + 1, high - 1)

return;

}



My Opinions

Thoughts, inspirations, and suggestions



村人登集祈雨，獨有一個孩子帶著雨傘前來。
這就是信心！

-佚名

9

My Notes

Important Concepts worth keeping

Towers of Hanoi

Today:

include <iostream>

using namespace std;

int step = 0;

void hanoi (int n, char A, char B, char C) {

if ($n == 1$) cout << A << "to" << C << '\n';

else {

hanoi (n - 1, A, C, B);

hanoi (1, A, B, C);

hanoi (n - 1, B, A, C);

}

}

from by to

A B C

from to by

n from by to

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

from by to

A B C

EVERY night we go to bed, without any assurance of being alive the next morning, but still we set the alarms.
That's HOPE!

-Anonymous

10

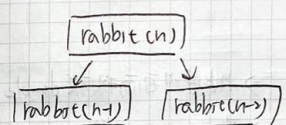
My Questions

Problems & Difficulties needing exploration

Multiplying Rabbits

* rabbit(n) = rabbit(n-1) + rabbit(n-2) 兔繁殖的數量

rabbit(2) = 1, rabbit(1) = 1



1st = 1 pair
2nd = 1 pair
3rd = 2 pairs
4th = 3 pairs
5th = 5 pairs
6th = 8 pairs

* Base case - rabbit(1), rabbit(2)

兔的定義 - rabbit(n) = 1 = rabbit(n-1) + rabbit(n-2) if $n > 2$

或 Base case - rabbit(1), rabbit(0)

= 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

My Opinions

Thoughts, inspirations, and suggestions

int rabbit (int n) {

if ($n <= 2$) return 1;

else // $n > 2$, so $(n-1) > 0$, $(n-2) > 0$ return rabbit(n-1) + rabbit(n-2);

} // 無效率

兔的繁殖次數?

rabbit(n) = 1 if $n <= 2$

= rabbit(n-1) + rabbit(n-2) + 1

= rabbit(n-1) + 1

= 9 次

n=5

rabbit(4)=3

rabbit(3)=2

return 5

n=4

rabbit(3)=2

rabbit(2)=1

return 3

n=3

rabbit(2)=1

rabbit(1)=1

return 2

n=2

return

return 1

return 1



我們仍然需要設定密碼。
這就是希望！

-佚名

11

My Notes

Important Concepts worth keeping

Today: / /

* 前題的遞迴 calls 次數

$$- n_1 = 1$$

$$- n_2 = 1$$

$$- n_3 = n_2 + n_1 + 1 = 1 + 1 + 1 = 3$$

$$- n_4 = n_3 + n_2 + 1 = 3 + 1 + 1 = 5 \Rightarrow n_k \text{ 呼叫每兩數就 double}$$

$$- n_5 = n_4 + n_3 + 1 = 5 + 3 + 1 = 9 \Rightarrow \text{That is, } n_k \geq 2^{k/2}$$

$$- n_6 = n_5 + n_4 + 1 = 9 + 5 + 1 = 15$$

$$- n_7 = n_6 + n_5 + 1 = 15 + 9 + 1 = 25 \Rightarrow \text{呼叫以指數成長 (bad)}$$

* Use linear recursion (cf)

Algorithm linearFibonacci(k) {

Input: integer k

Output: Pair of Fibonacci numbers (F_k, F_{k-1})

if k=1, then return (k, 0) // base case: k=1

else { $\Rightarrow (F_1, F_0)$

(i, j) = linearFibonacci(k-1) // (F_{k-1}, F_{k-2})

return (i+j, i) // (F_k = F_{k-1} + F_{k-2}, F_{k-1})

} 呼叫以線性成長 (k 次) (good)

WE see the world suffering, but still we get married.

That's LOVE!

-Anonymous

12

My Questions

Problems & Difficulties needing exploration

心得: 這門課第一堂的內容是遞迴, 雖然大一計概有學過一些了, 但當時使用卻完全不瞭解是如何運作的。這個章節我花了很多時間唸和看上課影片, 抄下了很多例子, 較知道如何運作了。但雖然遞迴的程式簡潔明確且節省記憶體空間, 可是它使用時參數的堆疊存取較費時, 且較不容易找到程式中哪裡出了錯誤, 必須要更熟悉遞迴才行。

My Opinions

Thoughts, inspirations, and suggestions



即便世態艱難, 我們仍然選擇陪伴同行。

這就是愛!

-佚名

13

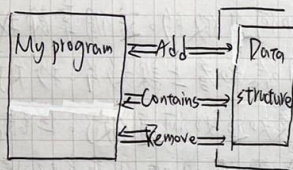
My Notes

Important Concepts worth keeping

抽象化

Today: / /

Abstract Data Type (ADT)



ADT = \emptyset A collection of data
 \emptyset A set of operations on that data

Specifications 描述 \Leftrightarrow Implementation 實作

* Inheritance 繼承: 在某種情況下, 一個類別會有「子類別」, 子類別比原本的類別 (父類別) 要更加具化, 也就是子類別繼承了父類別

* Overloading 多載: 在相同類別中, 定義名稱相同, 但是參數個數不同, 或是參數型態不同的函式。

To accomplish great things, we must not only act, but also dream, not only plan, but also believe.

-Anatole France

14

My Questions

Problems & Difficulties needing exploration

多載 Ex1:

class Rational {

public:

Rational add (Rational);

Rational add (long);

}

Rational Rational::add (Rational r) ...

Rational Rational::add (long i)

My Opinions

Thoughts, inspirations, and suggestions

心得: 雖然現在 C 語言沒有實作 ADT 的明確方法, 但上了這堂課之後知道了要怎麼自己設計, 而且使用者又需操作其所提供的函式, 非常方便, 之後可自己多練習。



為成就大事, 我們需要的不只是行動, 還要夢想, 不只是計畫, 還要信心。

-阿諾托爾·法郎士

15

My Notes

Important Concepts worth keeping

鏈串列

Today: / /

* pointer 包括 location or address in memory, of a memory cell.

* 表示 p 指向的 memory cell

$p = \&X$ 表示取出 X 的記憶體位置

$p = \text{new int}$ 表示多產生一個記憶體空間

$\text{delete } p$ 表示刪除 p 指標

$p = \text{NULL}$ 表示 p 沒指向任何東西

(Pointer - Based Linked List)

* A node in a linked list is usually a struct

Ex: struct Node {
int item;
Node *next;
}

item | next

* The head pointer points to the first node in a linked list

16. Only faith can let the thought spark, only hope can light up the future.
- Victor Hugo

My Questions

Problems & Difficulties needing exploration

* 如果 head 是 NULL, linked list 會是空的

心得: 第一次大一計概學指標時, 覺得非常抽象且無法瞭解概念。後來為我又有了更好的。但直到這堂課我才知道為何要使用 linked list, 且知道各種不同型態的 linked list, 像環狀 linked list, Dummy linked list, 双向 linked list 等。

My Opinions

Thoughts, inspirations, and suggestions



只有信心才讓思想發出火花, 只有希望才讓未來發出光芒。

- 蘋果 17

My Notes

Important Concepts worth keeping

定義語言

Today: / /

* 運算式 \Rightarrow Infix, Prefix, Postfix

If E is a prefix expression

if Y is any nonempty string of nonblanks

$\Rightarrow E \cdot Y$ cannot be prefix

* Backtracking 回溯

\Rightarrow 嘗試各種可能的解, 發現不通再退回來

心得: 這個章節老師給出了許多的例子去解釋各個不同的觀念, 讓我對於程式的瞭解度更上了一層。