My Notes

Important Concepts worth keeping

Today: / /

Recursive 遞迴.

A binary search is recursive

- Uses a divide and conquer strategy 分而擊之

Recursive Solution

1. Define the problem in terms of smaller problems
2. See if a recursive call decreases the problem size
3. Find a complete set of bases cases
4. Every time it will always reach a base case

1. 遞迴定義 sum(a~b) = sum(a+1~b+1) + b
2. 問題簡化 b → b+1
3. 終止條件 b, b+1, ~ a → b == a
4. 保證終止 natural numbers a > b

Linear recursion

```
int sum(int a, int b){
  //assume a<b
 if (a==b)
  return a;
 return a+sum(a+1, b);
}//sum()
```

Binary recursion

```
int sumB(int a, int n) {
  // assume n=(b)-(a+1)
  *  a1
 if (n==1)
  return n;
 return sumB(a, n/2) + sumB(a+n/2, n-n/2);
}//sumB()
```

Linear 比 Binary 更有效率.

人生不應像孤島，
有時轉身分享，
有時求助，
有時轉身打氣，
有時互相提攜。

My Questions
Problems & Difficulties needing exploration

My learning weather report

n_k be the number of recursive calls, $n_k \geq 2^{k/2}$ 呼叫次數以指數成長

費氏數列

抽象化

Encapsulation, 封裝 Inheritance, 繼承 Polymorphism, 多型.

⇒ hide inner details ⇒ reused. ⇒ object can decermine appropiate operations at execution time.

Modularity, 模組化.

A modularized program is, 易寫、易讀、易更改.

Isolates errors

highly cohesive modules desired, 高內聚 ⇒ 儘可能讓一模組只做一件事.

Loosely coupled modules desired, 低耦合 ⇒ 讓參數儘可能少量傳遞.

Data abstraction 資料抽象化: Asks you to think what you can do to a collection of data inpendentally of how you do it

My Opinions
Thoughts, inspirations, and suggestions

Specifications of an ADT indicate, 先分析描述問題

⇩

Implementation of an ADT, 實作出來. 使用程式語言.

each item has a unique predecessor and a unique successor

前進者 後繼者

列表中, 只有頭尾沒有.

insert (位置, List, Boolean) 插入 如位置有 data, 往後移.

remove (位置, Boolean) 刪除

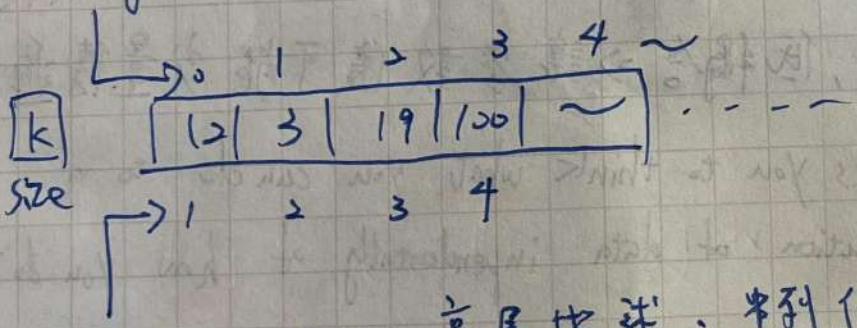
retriave (位置, List, Boolean) 檢索

replace => 先 remove 再 insert.

A class can have several constructors
跟記憶体要空間存資料.

Inheritance
An intance of a derived class can invoke public methods
子類別
of the base class
父類別

Array indexes 底層實作：陣到索引



ADT list positions 高層描述：串到位置.

ADT Polynomial
1. 多項者最高次項的指數
2. Power 次項的係數.
3. 將 Power 次項的係數改寫為 NewCoefficient
多項式各項的指數均為非負整數.

一步功，真輕鬆，
一步一腳印，
穩定中前進，
一點一滴滴，
日久見奇蹟。

My Questions
Problems & Difficulties needing exploration

My learning weather report

Array has a fixed size, 需要移动資料. 陣列

Linked list is able to grow in size as needed.
不需要移动資料. 鏈結串列.

int *p; // Initially undefined, but not NULL
// 一般變數. 直接配给.

p = &x; // the address-of operator &

p = new int; // the new operator, 緊急取置

Dynamic allocation of a memory cell that can contain an integer.

std::bad_alloc (in the <new> header)
沒有空房.

My Opinions
Thoughts, inspirations, and suggestions

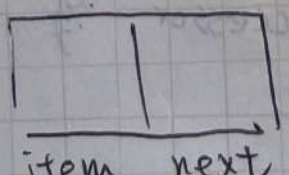
delete p; // 徹底清空.
p = NULL;

int arraySize = 50;

double *anArray = new double[arraySize]; // 动態陣列(配置)

Pointer-Based Linked Lists

int item
Node *next;



密碼 cipher-ke

先從終點想回頭，
再從起點開步走。
【終始功】

My Questions
Problems & Difficulties needing exploration

My learning weather report

Array has a fixed size, 需要移动資料. 陣列

Linked list is able to grow in size as needed.
不需要移动資料. 鏈結串列.

int *p; // Initially undefined, but not NULL
// 一般變數. 直接配给.

p = &x; // the address-of operator &

p = new int; // the new operator, 緊急取置

Dynamic allocation of a memory cell that can contain an integer.

std::bad_alloc (in the <new> header)
沒有空房.

My Opinions
Thoughts, inspirations, and suggestions

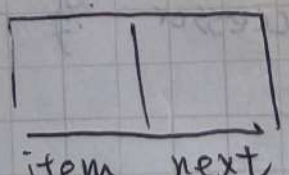
delete p; // 徹底清空.
p = NULL;

int arraySize = 50;

double *anArray = new double[arraySize]; // 动態陣列(配置)

Pointer-Based Linked Lists

int item
Node *next;



密碼 cipher-ke

先從終點想回頭，
再從起點開步走。
【終始功】