Date . .

The ADT Sorted List

Operation Contract for the ADT Sorted List

SortedIsEmpty(): boolean {query} 是為空

SortedGetLength(): interger {query} 計算個數

SortedInsert (in newItem: ListItemType, out success: bodean) 新增

SortedRemove (in index: interger, out dataItem: ListItemType, out success: boolean) {query} 檢索

locatePosition (in anItem: ListItemType, out isPresent: boolean): interger {query} 定位

Date . .

- A module's operation contract specifies its

1, purpose what actions take place?

2, Assumptions what does the module assume?

3, Input what data is available to a module?

4, Output what effect does the module have on the data

- Key Issues in Programming

1, Modularity 模組化

2, Style

3, Modifiability

4, Ease of Use

5, Fail-safe programming

6, Debugging

7, testing

Inheritance

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

Polymorphism

- object can determine appropriate operations at execution time. 多型

Operation Contracts 運算合約

- Document the use and limitations of a method
- Specify data flow
- Do not specify how module will perform its task
- Specify pre- and post-conditions

Date . .

Data Abstraction

描述 ADT 運算在前，實作在後

A class combines

1, Attributes (characteristics) of objects of a single type
- Typically data
- Called data members 屬性

2, Behaviors (operations)
- Typically operate on the data 運算
- Called methods or member functions

Principles of Object-Oriented Programming

- Three characteristics

✗ Encapsulation 封裝
- objects combine data and operations
- Hides inner details

→

Date

Greatest Common Diviser

problem
- Compute the GCD of two 非負整數 x, y

A recursive definition of GCD

$$gcd1(x,y) = x \quad \text{if } y=0$$
$$= gcd1(x, y \bmod x) \quad \text{if } y>x$$
$$= gcd1(y, x \bmod y) \quad \text{otherwise}$$

Another: $gcd2(x,y) = y$ if $x \bmod y = 0$

$= gcd(y, x, mody)$ otherwise

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

```
int gcd1 (intx, inty)
{
    if (y==0)    return x;
    elseif (y>x) return gcd1(x,y
    elle         return gcd1(x, x%y)
}
```

Date . .

Recursive Solution

1, 遞迴定義 $sum(a+....b) = sum(a+....b+1)+b$

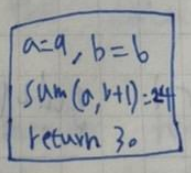
2, 問題簡化 $b \to b+1$

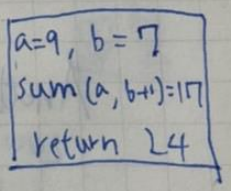
3, 終止條件 $b, b+1, ...a \to b == a$

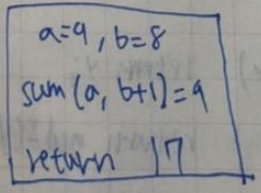
4, 保證終止 natural numbers $a > b$

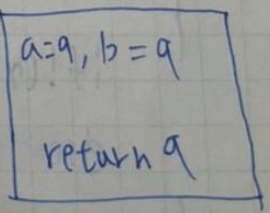
Recursive function

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



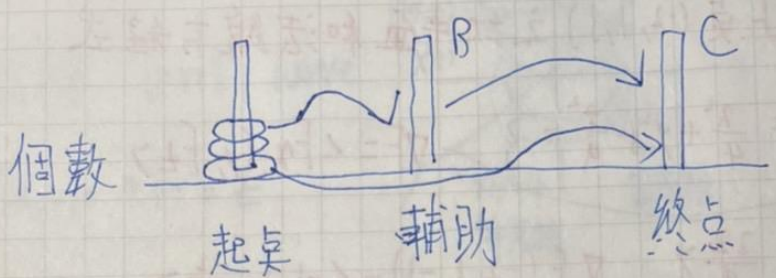






Date . .

Tower of Hanoi (河內塔)



$f(n) = f(n-1) + \cdots\cdots$

Writing a String Backward

```
Void writeBackward(string s, int size)
{
    if (size>0)
    {//write the last character
     cout << s.substr(size-1,1);   輸出最後一個字元
    //write the rest of the String backward
    writeBackward(s, size-1); //point A ⇒遞迴
    //size==0 → do nothing
} // End ....
```