My Notes

Important Concepts worth keeping

Today: / /

Choosing k out of n Thing

- $c(n,k) = c(n-1,k-1) + c(n-1,k)$

  取走一个 / 不取此项，往下看

* Solution

$$c(n,k) = \begin{cases} 1 & \text{if } k=0 \\ 1 & \text{if } k=n \\ 0 & \text{if } k>0 \\ c(n-1,k-1)+c(n-1,k) & \text{if } 0<k<n \end{cases}$$

递回呼叫次数 $c(n,k)-1$

Binary Search with an Array

```
int binarySearch(const int anArray[], int first, int last, int target){
    int index;
    if (first > last) index = -1;
    else {
        int mid = first + (last - first)/2;
        if (target == anArray[mid]) index = mid;
        else if (target < anArray[mid])
            index = binarySearch(anArray, first, mid-1,
                                 target);
        else
            index = binarySearch(anArray, mid+1, last,
                                 target);
    }// else
    return index;
}// end
```

Punctuality: Showing esteem for others by doing the right things at the right time.

My Questions
Problems & Difficulties needing exploration

Binary Recursion

- 若非遇到 Base case，則每次會再次呼叫兩個遞迴.

費伯納西數列

關係式：rabbit(n) = rabbit(n-1) + rabbit(n-2) ◎每月兔子數為前兩月相加
rabbit(2) = 1 ◎第二個月只有一對兩個月大的
rabbit(1) = 1 ◎第一個月只有一對

定義1：Base Case – rabbit(2), rabbit(1)

Recursive rabbit(n) = 1 if n = 1 or 2
= rabbit(n-1) + rabbit(n-2) if n>2

定義2：Base Case – rabbit(1), rabbit(0)

Recursive rabbit(n) = n if n = 1 or 0
= rabbit(n-1) + rabbit(n-2) if n>1

遞迴呼叫次數（類似費伯納西數列）

Ex: rabbit#(7) = rabbit#(6) + rabbit#(5) + 1
自己呼叫

My Opinions
Thoughts, inspirations, and suggestions

Organizing a Parade

P(n) 方法總數
F(n) 隊伍最後是 float 的數量
B(n) 隊伍最後是 band 的數量
n: 隊伍總長

P(n) = F(n) + B(n)

* Solution
P(1) = 2
P(2) = 3
P(n) = P(n-1) + P(n-2)
for n > 2

My Notes

Important Concepts worth keeping

Today: , ,

河內塔:

```
solveTowers(count, source, destination, spare)
    if(count is 1) Move a disk directly from source
                                to destination
    else{
        solveTowers(count-1, source, spare, destination)
        solverTowers(1, source, destination, spare)
        SolverTowers(count-1, spare, destination, spare)
    }//else
```

移動次數: 2^n-1 (n: 幾層)

Binary Recursion

- 若非遇到 Base Case, 則每次會再次呼叫兩個遞迴.

Constructors

- 新增並初始化 instances (invoked when declare an instance)
- 和 class 同名
- 沒有 return 值, void 也不是.
- 可以同時有多個 constructor, default 是沒有參數
- compiler 會幫你做一個

My Questions
Problems & Difficulties needing exploration

Destructors
- destory instance when it's lifetime ends.
- Compiler also generate one

Encapsulation 封裝
```
class ex {
    :
};
```
封裝為特程式碼隱藏起來只讓使用者只使用其 function

Inheritance 繼承
```
class ex : public exsuper
{
    :
};
```
ex 為繼承 exsuper 此父類別的子類別

Overloading 重載
```
void anExample(int num)
void anExample(char ch)
```
提供統一名稱，但根據參數列個數或資料型態不同，而自動呼叫對應函式

Overriding
```
class Person {
    public void Print() {
        cout << "I'm a person";
    } //Print()
};
```

```
class Superman : public Person {
    public void Print() {
        cout << "superman";
    } print()
}
```

子類別將父類別函式重新定義以符合自身所需

My Notes

Important Concepts worth keeping

Today: , ,

Call by value 傳值.

```
void swap (int c, int d) {
    int temp = c;
    c = d
    d = temp;
}
int main() {
    int a=1 b=10;
    swap (a, b);
}
```

只把值拿來暫時存到 c, d (新的記憶體位置) 原本 a, b 不改

Call by address 傳址.

```
void swap (int *c, int *d) {
    int temp = *c;
    *c = *d;
    *d = temp;
}
int main() {
    int a=1, b=10;
    swap (&1, &b);
}
```

和 call by value 很像, 只是該 value 的值剛好是 a, b 的記憶體位址

My Questions
Problems & Difficulties needing exploration

Call by reference 傳參考

```
void swap ( int &c, int &d){
    int temp = c;
    c=d
    d= temp;
}
int main() {
    int a=5, b=10;
    swap (a, b);
}
```

傳參考可說是變數的別名,綽號
* 若一變數已是另一變數的參考則不能再參考他人,且傳參考值必須在一開始宣告就給參考對象,無法之後再給

My Opinions
Thoughts, inspirations, and suggestions

My Notes

Important Concepts worth keeping

Today: / /

八皇后問題

* 將八个皇后放入64格的棋盤中，並令他們不會互相攻擊

· **Base Case** if there are no more columns to consider (finish!)

Recursive step if (成功將一皇后放進一column)
繼續看下一column
else if (目前column都不能放) 回溯

```
Void swapFirstLast (int aList, out success)
    aList.retrieve (1, firstItem, success);
    aList.retrieve (aList.getLength(), lastItem, success);
    aList.remove (1, success);
    aList.remove (aList.getLength(), success);
    aList.insert (1, lastItem, success);
    aList.insert (aList.getlength()+1, firstItem, success);
Void reverseElementOrder (int aList, out success)
    for (int i=1; i< aList.getLength(); i++)
    {
        aList.retrieve (1, dataItem, success);
        aList.remove (1, success);
        aList.insert (aList.getLength()-i+2, dataItem,
                      success);
    }// end for
```


Let your Yes be Yes, and your No be No.
《Mathew》

My Questions
Problems & Difficulties needing exploration

Pointer-Based 和 Array-Based 鏈結串列

[Size] Array 若要新增空間比 pointer 耗時耗空間

[Storage Require] Array 佔用的記憶體空間較 Pointer 少

[Retrieval] Array 可以直接找到第 i 個 item, Pointer 要移動 i 次

[Insertion, Deletion] Array 移動全部資料, Pointer 移動部分

[Ordering Scheme] Array 隱性排列順序, Pointer 明顯先後

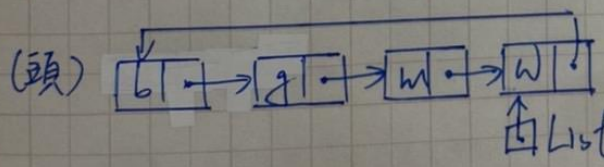
[Access Element] Array → directly, Pointer → traversal

Circular Linked List

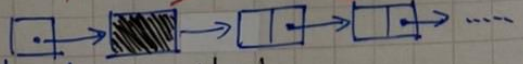
- Last node points to the first node
- Every node has a successor (繼承者)
- No node in circular linked list has NULL

My Opinions
Thoughts, inspirations, and suggestions

- Access to Last node require traversal
- 令設一指標指向尾端可同時存到頭和尾的資料 Cavoid traversal



Dummy Head Node



可直接用 dummy head delete / insert，而不用改 head，dummy head 一直存在。

My Notes

Important Concepts worth keeping

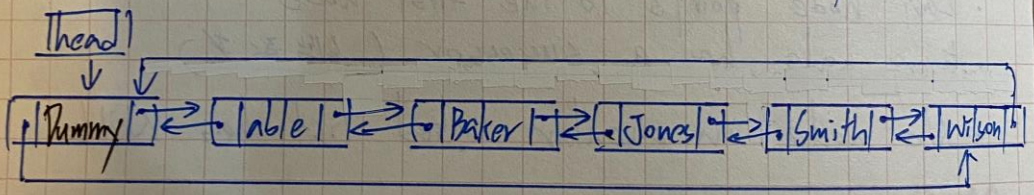
Today : / /

Doubly Linked Lists

- Each node points to its predecessor (前者) and its successor (後者)
- Often has a dummy head
- Often circular to eliminate special cases

```
struct Node {
    int item
    Node *precede
    Node *next
};  // end Node
```

* 有 dummy head node：
dummy head 的 precede 指向尾端，尾端的 next 指向 dummy head



My Questions
Problems & Difficulties needing exploration

算術運算式

原始 (a+(b*c))

後序運算式 (postfix)：運算子都放到對應的右括號')'
*若式子是完全括號的

前序運算式 (prefix)：運算子都放到對應的左括號'('

定義語言

<identifier> = <letter> | <identifier> | <identifier> <digit>
遞迴方式運作，可產生無限長字串

Ex: <S> = <U> | <S> <D>
<D> = 1 | 0
<U> = A | B

Q1: Write all string in language
A1: A11, A10, A01, A00, B11, B10, B01
Q2: Is AB001 in language?
A2: No, 尾巴多一個 1

My Opinions
Thoughts, inspirations, and suggestions

Q3: Modify above grammer to define a language of bit-strings that first character is 1 and the last is 0

A3: <S> = 10 | 1<X>0 |
<X> = 0 | 1 | 0<X> | 1<X> |