

《程式語言》

試題評析

今年程式語言考題除了第四題外其餘都是經典題型的變化，題目均是重要章節的主要重點。分析如下：

第一題：是標準遞迴程式的範圍。是Fibonacci數列的變化。要注意的是題目規定答案複雜度不得高於 $O(n)$ ，代表作答時要以反覆(Iterative)技術實做此題遞迴問題。

第二題：是抽象資料型態的實做，要注意的是題目已經給定實做函式的函式原型(Prototype)，同學必須以題目規定的原型來思考，用鏈節串列實做佇列的方法。

第三題：是基本文法題，屬於模糊文法(Ambiguous grammar)中運算式例子的變形，同學只要分析運算式運算子優先順序以及結合律就可以輕鬆解答此題。

第四題：屬於物件導向概念章節的部分，雖然較偏系統分析的範疇，但是題目要求畫出的類別圖只用到簡單的繼承、包含及相關等關係，稍微熟悉UML即可完成此題。

綜合來說今年題目不難，一般程度學生可以拿65~80分，程式寫作基本功紮實的同學要拿80分以上不是難事。

一、我們有下列的C/C++語言程序。

```
void f(n)int n;{
    if(n==1)printf( "*" );
    else if(n==2)printf( "***" );
    else {f(n-1); f(n-2); f(n-2);}
}
```

假設 $a(n)$ 是 $f(n)$ 列印出的星號數目。請寫出一個程式，可以算出 $a(n)$ 。這程式必須具有不可高於 $O(n)$ 的複雜度。此外，請列出 $a(n)$ 的遞迴方程式(recurrence equation)，並寫出其解答。(25分)

答：

```
int a(int n)
{
    if (1 == n) {

        return 1;
    } else if (2 == n) {

        return 3;
    } else {
        int item1 = 1;
        int item2 = 3;
        int item3 = 0;
        for (int i = 3; i <= n; i++) {

            item3 = item2 + 2*item1;
            item1 = item2;
            item2 = item3;
        }
        return item3;
    }
}
```



```

}
}

```

二、假設我們有一個下列的C/C++語言的資料結構。

```
struct node_type{int value; struct node_type *next; };
```

請用這個資料結構，設計一個queue的enqueue與dequeue程序。

```
struct node_type *enqueue(struct node_type *q, int v);
```

這個程序會把v加到q指到的queue的尾端，並且把新的queue傳回。

```
struct node_type *dequeue(struct node_type *q, int *vp);
```

這個程序會把q指到的queue頭的值，寫到*vp中，然後把這個頭去掉，再把新的queue傳回。這個程序的complexity都必須是 $O(1)$ ，而且必須能處理空的queue。你只能使用一個static pointer變數。(25分)

答：

```

struct node_type {
    int value;
    node_type *next;
};
node_type* enqueue(node_type *q, int V) // add V to tail, return new queue
{
    static node_type *pQueueTail;//此靜態指標變數用來記錄佇列的尾巴
    node_type *pNewNode = new node_type;
    pNewNode->value = V;
    pNewNode->next = NULL;

    if (NULL == q) {
        //當佇列為空時直接將新產生的佇列節點當作新佇列的開頭
        pQueueTail = pNewNode;
        return pNewNode;
    } else {
        //當佇列已經存在元素時，將新元素插入佇列尾端
        pQueueTail->next = pNewNode;
        pQueueTail = pNewNode;
        return q;
    }
}

node_type* dequeue(node_type *q, int *vp)//write first to *vp, remove head, return new queue
{
    if (NULL == q) {
        //傳回NULL值表示佇列為空
        *vp = 0;
        return NULL;
    }
}

```

```

} else {
    //佇列非空時將開頭元素資料傳回，並且釋放此元素所佔用的記憶體
    *vp = q->value;
    node_type *pNewListHead = q->next;
    delete q;
    return pNewListHead;
}
}

void main(){
    node_type* front = NULL;//此代表佇列的頭(front)
    //將10, 20插入佇列中
    front = enqueue(front, 10);
    front = enqueue(front, 20);
    int value = 0;
    取出剛剛插入佇列的元素
    front = dequeue(front, &value);
    cout << "value is:" << value << endl;
    front = dequeue(front, &value);
    cout << "value is:" << value << endl;
}

```

三、我們有下列的BNF(Backus Naur Form)文法：

$$S \rightarrow a \mid S+S \mid S-S \mid M$$

$$M \rightarrow a \mid M*M \mid M/M \mid (S)$$

請問這個文法，是不是ambiguous。如果是的話，可以舉出一個例子嗎？如果不是的話，請證明為什麼它不是ambiguous。（25分）

答：

此文法為模糊文法(ambiguous grammar)，因為運算式的結合性未定義。

例如：

$a + a - a$ 可以有以下兩種剖析樹(parsing tree)

1.

```

      S
     / | \
    S + S
   / \ / | \
  a  S - S
     |   |
     a   a

```

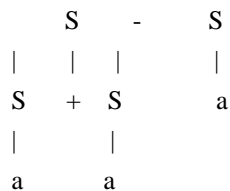
2.

```

          S
         / | \
        | | |

```





四、UML是Unified Modeling Language的簡寫。請畫出一個家庭class（類別）的UML class diagram。這一個家庭class，要含有下列的特性(properties)：（25分）

- 家庭姓氏、家長姓名、地址、家庭電話。
- 家庭成員的資訊，包含了其姓名、性別、出生年月日、已婚還是未婚、與家長的關係。
- 每個家庭成員若是已婚子女，還要有pointer指到他現在與配偶組成的家庭object。
- 我們要有methods，可以增加家庭成員，可以刪除家庭成員，可以更改家長姓名，可以改變子女的婚姻狀態。
- 另外，我們還希望有家庭的subclasses（次類別），其中一種是低收入戶家庭，我們要記載他們每月的政府補貼金額。還有非低收入戶家庭，要記錄他們家的每年平均稅率。
- 如果是低收入戶家庭，我們希望能夠有methods，可以查詢他們的每月政府補貼金額。
- 如果是非低收入戶家庭，我們也要有methods，可以在經濟不景氣時，申請成為低收入戶家庭。

答：

