

# 《程式設計概要》

一、請試述下列名詞之意涵：（每小題4分，共24分）

- (一)AVL tree
- (二)Merge Sort
- (三)BNF Grammar
- (四)Dynamic Scope
- (五)Dynamic Linker
- (六)Heap-dynamic Variable

**試題評析** 本題主要測試程式語言與資料結構的相關名詞，對於平常有注意觀念的學員可輕鬆取分。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁123。

**答：**

(一)AVL tree：高度平衡樹，有下列特性

- 1.是一棵二元搜尋樹(Binary Search Tree)
- 2.對任一個節點，左子樹與右子樹高度差需在1以內。

(二)Merge Sort：將兩個已排序的陣列合併到一個已排序新的陣列中。

(三)BNF Grammar：是一種用於表示上下文無關文法的語言，上下文無關文法描述了一類形式語言，它是由約翰·巴科斯(John Backus)和彼得·諾爾(Peter Naur)首先引入的用來描述計算機語言語法的符號集。

(四)Dynamic Scope：一個變數的範圍(scope)，就是指它可以被引用的那些程式區段。最常採用的定義範疇規則有下列兩種：

- 1.靜態範圍(static scope)：主要是依程式本身，變數的實際位置來決定。
- 2.動態範圍(dynamic scope)：依程式執行時，副程式的呼叫順序來決定。

(五)Dynamic Linker：當需要使用到函數時，才尋找並連結函式庫，這種方式可以不用載入全部的程式，因此可以節省記憶體。當很多執行中的程式共用到函式庫時，動態連結所節省的記憶體數量就相當可觀。

(六)Heap-dynamic Variable：Heap是一塊可永久使用的記憶體，由使用者來決定其增減的，它主要存放所有"new"出來的dynamic variable，也就是在執行期間請求的記憶體。

二、int a; 則變數a最小值與最大值分別為何？（3分）

**試題評析** 本題是測試整數型態的基礎觀念。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁13。

**答：**

在C語言中，int佔4 Bytes(32 bits)。所以最大與最小值分別 $2^{31}-1$ 與 $-2^{31}$ 。

三、定義一個函數如下：

```
int f(int n) {if(n==0) return 0; if(n==1) return 1; if(n==2) return 2;
return f(n-1)+f(n-2)+f(n-3);}
```

請問計算f(6)時，共呼叫f(n)幾次？（8分）

**試題評析** 本題是測試遞迴的觀念，使用帶入法就可計算答案。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁125-133。

**答：**

f(0)、f(1)與f(2)是邊界條件，呼叫一次就結束。

f(3)呼叫f(0)、f(1)與f(2)，共4次

f(4)呼叫f(1)、f(2)與f(3)，共7次  
 f(5)呼叫f(2)、f(3)與f(4)，共13次  
 f(6)呼叫f(3)、f(4)與f(5)，共25次(包含f(6)這一次)

四、有C程式如下，請寫出函數add()正確的函數定義，輸入為一個二維array，輸出為一整數值。  
 (8分)

```
void main() { int a[12][15], sum;
              sum=add(a, 12, 15);}
```

**試題評析** 本題是雙重迴圈操作陣列的觀念，利用外圈控制列、內圈控制行就可以解決問題。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁95。

**答：**

```
int add(int a[][15], int row, int col){
    int i, j, total=0;
    for(i=0; i<row; i++){
        for(j=0; j<col; j++){
            total+=a[i][j];
        }
    }
    return total;
}
```

五、若有N個資料，每次做資料處理時都需選最大，請依下列資料結構：unordered linked list、sorted array及heap，分別寫出這些資料作insert及delete時的時間複雜度。(12分)

**試題評析** 本題是測試資料結構的觀念，使用鏈結串列、陣列與堆積觀念就可以輕鬆作答。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁92、176。

**答：**

對最大的資料進行新增(Insert)與刪除>Delete)所需的複雜度:

	Insert	Delete	備註
Unordered linked list	O(n)	O(n)	搜尋最大值就需要O(n)
Sorted array	O(1)	O(1)	最大值在最後一個位置
Heap	O(1)	O(1)	使用最大堆積樹，樹根就是最大值

六、假設有一個演算法，它的計算量可寫成如下的遞迴式 $T(n)=T(n-1)+1/n$ ， $T(1)=1$ ，請問此演算法的時間複雜度為何？(8分)

**試題評析** 本題是測試遞迴的觀念，使用帶入法就可計算答案。最後產生結果需要使用到簡單的積分觀念。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁125。

**答：**

$O(\log(n))$   
 $T(n)=T(n-1)+1/n$   
 $=T(n-2)+1/(n-1)+1/n$   
 $=\dots$   
 $=T(1)+1/2+1/3+\dots+1/(n-1)+1/n$   
 $=1/1+1/2+1/3+\dots+1/(n-1)+1/n$

【版權所有，重製必究！】

$$H(N) = 1 + 1/2 + 1/3 + \dots + 1/N$$

$$\int_1^{N+1} \frac{1}{t} dt \leq H_N \leq 1 + \int_1^N \frac{1}{t} dt.$$

$$\ln(N+1) \leq H(N) \leq 1 + \ln(N)$$

$$H(N) = \Theta(\log(N))$$

七、請用非遞迴的方式，寫出一副程式gcd(int m, int n)，藉以求出兩整數m與n之間的最大公因數。(8分)

**試題評析** 本題是測試迴圈的觀念，上課就有特別提醒是重要問題，果然完全命中。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁78。

**答：**

```
int gcd(int m, int n){
    int result, rem;

    while((rem=m%n)!=0){
        m = n;
        n = rem;
    }
    return n;
}
```

八、給定一個二元樹T，它的inorder sequence為“maxengbyc”；它的preorder sequence為“gamexncby”。

(一)請將T構建出來。(5分)

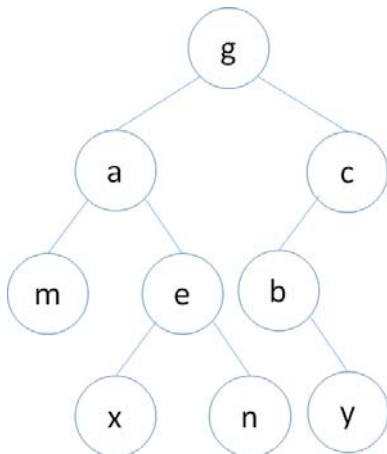
(二)為何只給preorder與postorder的結果，無法唯一決定出一棵二元樹？(3分)

**試題評析** 本題主要測試二元樹的觀念，有注意上課的學員可輕鬆取分。

**考點命中** 《高點·高上程式設計概要講義》第一回，許振明編撰，頁203。

**答：**

(一)



權所有，重製必究！】

(二)如果只給preorder與postorder，可能會有兩棵或兩棵以上的二元樹會符合條件，因此無法決定唯一的一棵二元樹。

九、請依下列程式求出x與y的值。(9分)

```
int x=0, y=0;
for(int i=0;i<100;i++)
  for(int j=i+1;j<100;j++)
    {x++;
     for(int z=j+1;z<=100;z++)y++;
    }
```

<b>試題評析</b>	本題是雙重迴圈操作陣列的觀念，利用外圈控制列、內圈控制行就可以解決問題。
<b>考點命中</b>	《高點·高上程式設計概要講義》第一回，許振明編撰，頁95。

**答：**

x:4950

y=166650

十、請用遞迴的方式，寫出quicksort(int \*A, 0, n-1)的副程式，利用quicksort的演算法，將A陣列裡的n筆資料，從小排到大。(12分)

<b>試題評析</b>	本題是排序的重要觀念，上課就有特別提醒是重要問題，果然完全命中。
<b>考點命中</b>	《高點·高上程式設計概要講義》第一回，許振明編撰，頁190。

**答：**

```
void quickSort(int *list, int left, int right){
    int pivot,i,j,temp;

    if(left<right){
        i=left;
        j=right+1;
        pivot=list[left];
        do{
            do{
                i++;
            }while(i<=right && list[i]<pivot);
            do{
                j--;
            }while(j>left && list[j]>pivot);
            if(i<j){
                temp=list[i];
                list[i]=list[j];
                list[j]=temp;
            }
        }while(i<j);

        temp=list[left];
        list[left]=list[j];
        list[j]=temp;
    }
```

【版權所有，重製必究！】

```
    quickSort(list,left,j-1);  
    quickSort(list,j+1,right);  
  }  
}
```

# 高點 · 高上

【版權所有，重製必究！】