《資料結構》

試題評析	本次檢察事務官之資料結構試題，比前幾次要來得不易作答，主要是應用變化題與程式撰寫題居多。即使應試者每題都能夠迅速掌握到要點，在時間限制之內要完全作答，恐亦非易事。但也因而更能測驗出應試者，對相關知識的熟悉程度與應變能力。此外，因為題目份量多，有些又具有難度，應試者的情緒，能否冷靜作答，可能反而是影響成績的重大因素。大致而言，若不考慮答題時間是否充足之問題，可算是相當不錯的命題。 各題的特點如下：第一題是鏈結串列的應用題，屬於程式設計基礎問題，主要在測驗應試者的鏈結串列處理能力與長整數加法的基本計算法。第二題為較基本的問題，二元搜尋樹基本操作與追蹤法，對應考者應該可以輕易拿分。第三題為二維陣列空間安排的變化題，難度不算難，但須冷靜作答，應可以拿到分數。第四題必須先想到要使用基數排序法，否則無法做到時間複雜度O(N)的要求，但仍須撰寫出程式，最重要的還是要「冷靜作答」。第五題為鏈結串列基本處理，要拿分應不難。第六題須先做遞迴關係式的分析，但後半段非遞迴程式撰寫部份，須使用動態程式設計法，否則較難寫出來。 整份試題除了第二題與第五題較基本之外，其餘問題，幾乎題題皆關鍵，多答對任何一題，都會為自己增加不少的上榜機會。預估本科目應考者的分數範圍應分佈在40~80之間。

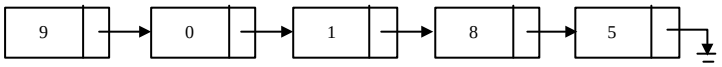
一、設計一種資料結構可以用來儲存任意大小的非負整數（包含零及正整數），並請寫一段程式在這資料結構上做兩非負整數的加法運算。（15分）

【擬答】

使用鏈結串列來儲存此一非負整數，其節點結構如下：

```
struct node
{    int digit;
     struct node * next;
};
typedef struct node * nodeptr;
```

例如：58109 可以建立如下之之鏈結串列，個位數字在最開頭，愈高位數字儲存在串列愈尾部



兩串列數字法運算如下：

```
nodeptr listadd (nodeptr p, nodeptr q)
{  nodeptr head, tail, r;
   int pdigit, qdigit;
   int carry=0, sum;
    head=(nodeptr)malloc(sizeof(struct node));
    tail=head;
   while (p!=NULL && q!=NULL)
   {  if (p==NULL) pdigit=0;
      else {    pdigit=p→digit;
                p=p->next;        }
     if (q==NULL) qdigit=0;
      else {    qdigit=q->digit;
                q=q->next;        }
       sum=pdigit+qdigit+carry;
       tail->next=(nodeptr)malloc(sizeof(struct node));
       tail=tail->next;
       tail->digit=sum%10;
       carry=sum/10;
    }
    tail->next=NULL;
    r=head->next;
    free(head);
   return r;
 }
```

二、假設一組數為：A=8，B=17，C=15，D=10，E=12，F=19，G=6，H=30，I=5，J=14**考慮以下序列運算：**

Insert(A)，Insert(B)，Insert(C)，Insert(D)，Insert(E)，Insert(F)，Insert(G)，Insert(H)，Insert(I)，Insert(J)，

Delete(A)。（20分）

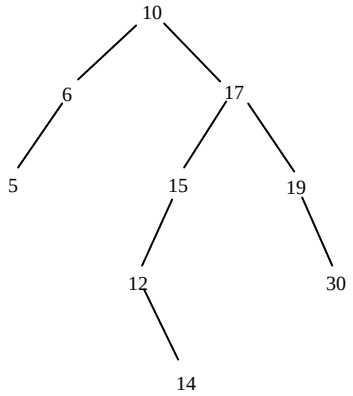
(一)請描述binary search tree的特性，並請畫出經過以上序列運算所建構的binary search tree。

(二)請寫出所建構之binary search tree的preorder，inorder，postorder，breadth-first order。

【擬答】

(一) Binary Tree 的特性：是一棵 binary tree，若其不是空的 binary tree，則樹根(root)的左子樹中的節點所儲存之資料，皆不大於樹根；而右子樹中的節點所儲存的資料，皆不小於樹根。而且、右子樹亦皆為 binary trees。

這組數所建立之最後 binary tree 如下：



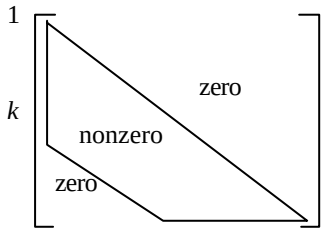
(二) preorder: 10,6,5,17,15,12,14,19,30
inorder:5,6,10,12,14,15,17,19,30
postorder:5,6,14,12,15,30,19,17,10
breadth-first order:10,6,17,5,15,19,12,30,14

三、一個n乘n的方陣A=[ai,j]，1≤i≤n，1≤j≤n，若存在一數k，k≤n，使得i–j≥k與i–j<0時ai,j=0；則此方陣稱為下三角斜條方陣（lower triangular banded matrix）。若0≤ i–j<k則稱元素ai,j，在斜條（band）上。

請設計一方法將斜條上的元素儲存至一維陣列D。（20分）

(一)問D總共有多少元素。

(二)假設0≤ i–j<k，請寫出ai,j儲存於陣列D的指標位置。



【擬答】

(一) 共有 $n+(n-1)+(n-2)+...+(n-k+1)=\frac{(2n-k+1)k}{2}=nk-\frac{k(k-1)}{2}$ 個元素。

(二) 以 row-major 方式來安排 non-zero 元素的空間即可，a11 置於 D[1]，a21 置於 D[2]，a22 置於 D[3]，a31 置於 D[4]，餘此類推。若 aij 置於 D[k]，則 k 與 i,j 之關係式如下:

$$k=\begin{cases}\frac{i(i-1)}{2} & ,1\le i\le k\\ \frac{k(k+1)}{2}+k(i-k)-i+j & ,k<i\le n\end{cases}$$

註：此題亦可採用對角線的次序，由最下方(或最上方)的對角線開始，一條一條向上(向下)逐一安排空間，則公式將有所不同。

四、假設有一組整數，每個數都介於0與31之間。請寫一個程式將這組數依小到大排列。若這組整數的個數為*N*，這程式所需要的計算次數最多為O(*N*)。（20分）

【擬答】

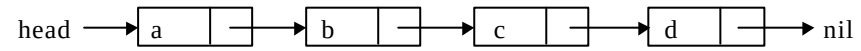
使用 radix=2 的 LSD(Least-Significant Digit) Bucket Sort 即可，假設N 個資料已建立並儲存於 single linked list 中，兩個 buckets(以linked lists表示) 皆各有首節點，並以 head[0],head[1]與tail[0],tail[1]分別指向開頭與結尾的節點。

```
nodeptr LSDBucketSort(nodeptr p)
{   nodeptr head[2], tail[2];
    int pass, m;
    nodeptr q;
    head[0] =(nodeptr)malloc(sizeof(struct node));
    head[1]=(nodeptr)malloc(sizeof(struct node));
   for (pass=1, m=1; pass<=5; pass++, m*=2)
    {   tail[0]=head[0];   tail[1]=head[1];
        while (p!=NULL)
        {   q=p;   p=p->next;
            i=(q->data & m) >> (pass-1);
            tail[i]->next=q;
            tail[i]=q;
        }
        tail[0]->next = tail[1]->next = NULL;
        if (head[0]==tail[0]) p=head[1]->next;
        else { p=head[0]->next; tail[0]->next=head[1]->next; }
    }
    return p;
}
```

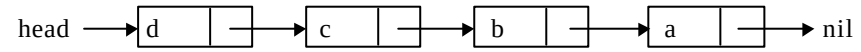
因為總共進行 5 passes 的運算，而每一回所需的時間皆為 O(N)，故總時間仍為 O(N)。

五、寫一程式將一個linked list的所有元素（entries）的次序完全倒過來（如下圖所示）。（10分）

原來的linked list：



次序完全倒過來的linked list：



【擬答】

```
nodeptr ListReverse(nodeptr head)
{   nodeptr reversed, temp;
    reversed=NULL;
    while (head!=NULL)
    {   temp=head;
        head=head->next;
        temp->next=reversed;
        reversed=temp;
    }
   return reversed;
}
```

六、令長度為n的序列集合

An={x1,x2,…,xn | xi=0,1**或**2，**對**1≤i≤n−1，xi,xi+1≠1,1**且**xi,xi+1≠2,2}（**即序列中沒有連續兩個1或是連續兩個2**）。

設f(n)**為**An**的個數，**f0(n), f1(n), f2(n)**分別為**An**中**x1=0,1,2**的序列的個數。請導出**f(n)**的遞迴公式（**recursive formula**），並請寫一非遞迴函式計算**f(n)。（15分）

【擬答】

遞迴公式為

$$f_k(n)=\begin{cases} f_0(n-1)+f_1(n-1)+f_2(n-1) & ,n>1\text{且}k=0 \\ f_0(n-1)+f_2(n-1) & ,n>1\text{且}k=1 \\ f_0(n-1)+f_1(n-1) & ,n>1\text{且}k=2 \\ 1 & ,n\le 1 \end{cases}$$

而 $f(n)=f_0(n)+f_1(n)+f_2(n)$

故可以採用 dynamic programming **方式來計算** $f_k(n)$**，並以一個** 2-D array a[k][m] **來儲存計算過的**$f_k(m)$**，其中**$1\leqq m\leqq n$**，**$0\leqq k\leqq 2$**程式如下：**

```
int f(int k, int n)
 {   int m;
     int a[3][MaxN];
      a[0][1]=a[1][1]=a[2][1]=1;
     for(m=2; m<=n; m++)
      {   a[0][m]=a[0][m-1]+a[1][m-1]+a[2][m-1];
          a[1][m]=a[0][m-1]+a[2][m-1];
          a[2][m]=a[0][m-1]+a[1][m-1];
      }
     return a[0][n]+a[1][n]+a[2][n];
 }
```