大碩教育出版有限公司

# 研究所講重點【資料結構(含精選試題)】

# 【書籍勘誤表】

(書編：AE60160701，於 2021 年 9 月修訂)

P. 53

## 原錯誤處

試題 37

Let goo(m) be a function with runtime complexity O(m) for the parameter m. What is the runtime complexity of the following function foo(n)?

```
void foo(int n) {
    for (int  i = 0;  i < n;  i++) {
        for (int  j = 1;  j >= 0;  j = j / 2 )
            goo (j) ;
    }
}
```

(A) O(n) (B) O(n log n) (C) $O(n^2)$ (D) $O(n^2 \log n)$

(E) $O(n^3)$ 【101 台大電機資料結構】

## 需修正為

試題 37

Let goo(m) be a function with runtime complexity O(m) for the parameter m. What is the runtime complexity of the following function foo(n)?

```
void foo(int n) {
    for (int  i = 0;  i < n;  i++) {
        for (int  j = i;  j > 0;  j = j / 2 )
            goo (j) ;
    }
}
```

(A) O(n) (B) O(n log n) (C) $O(n^2)$ (D) $O(n^2 \log n)$

(E) $O(n^3)$ 【101 台大電機資料結構】

P. 55-56

## 原錯誤處

試題 42

We wrote a recursive function fab to compute Fibonacci numbers, and we used a global array count to store the number of times fab(n) was called. Please predict the output of this program, and give detailed reasoning for your answers. (15%)

```
#include <stdio. h>
#define N 10

int count [N + 1];

int fab (int n)
{
     /* Keep track the number of times fab(n) was called */

     count[n] ++
     if (n = = 0 || n = = 1)
        return 1;
     else
        return (fab(n – 1) + fab(n – 2));
}
int main ( )
{
     int 1;
     /* Although global variables are initialized automatically, we still do this to avoid confusion. */

     for (i = 0; i < = N; i ++)
        count[i] = 0
     fab(N);

     /* Print the values of count [ ]. */

     for (i = 0; i < = N; i ++)
        printf (“count [%d] = %d\n”, i, count [i];
}
```

【99 台大資工軟設】

## 需修正為

試題 42

We wrote a recursive function fab to compute Fibonacci numbers, and we used a global array count to store the number of times fab(n) was called. Please predict the output of this program, and give detailed reasoning for your answers. (15%)

```
#include <stdio. h>
#define N 10

int count [N + 1];

int fab (int n)
{
     /* Keep track the number of times fab(n) was called */

     count[n] ++
        if (n = = 0 || n = = 1)
            return 1;
        else
            return (fab(n – 1) + fab(n – 2));
}
int main ( )
{
        int i
        /* Although global variables are initialized automatically, we still do this to avoid confusion. */
        for (i = 0; i < = N; i ++)
            count[i] = 0
        fab(N);
        /* Print the values of count [ ]. */
        for (i = 0; i < = N; i ++)
            printf (“count [%d] = %d\n”, i, count [i]);
}
```

【99 台大資工軟設】

| P. 96 | 原錯誤處 |
|---|---|
|  | (三)將算術中之中序表示法[Infrix(LDR)]轉換成後序[PostFix(LRD)]或前序式表示法以便執行算術之計算。後序式計算及前序式計算也是 stack 應用。 |
|  | 需修正為 |
|  | (三)將算術中之中序表示法[Infix(LDR)]轉換成後序[PostFix(LRD)]或前序式表示法以便執行算術之計算。後序式計算及前序式計算也是 stack 應用。 |

| P. 124 | 原錯誤處 |
|---|---|
| | 試題 29 解答<br>ANS<br>(1) full: $(start + 1)\%n == end$，n 為 Queue size<br>empty: $statr == end$<br>(2) full: (start == end) && (Tag == True)<br>empty: $(start == end)\ \ \&\&(Tag == False)$ |
| | **需修正為** |
| | 試題 29 解答<br>ANS<br>(1) full: $(end + 1)\%n == start$，n 為 Queue size<br>empty: $statr == end$<br>(2) full: $(start == end)\ \ \&\&(Tag == True)$<br>empty: $(start == end)\ \ \&\&(Tag == False)$ |

| P.198 | 原錯誤處 |
|---|---|
| | 試題 3<br>Which of the following is (are) true about the “tree” data structure?<br>(A) Number of internal nodes = number of external (leaf) nodes – 1,<br>(B) Number of edges = number of nodes – 1,<br>(C) For each external (leaf) node, there exists only one trace from the root node to this external node,<br>(D) Each node in the tree can be a root node of a subtree,<br>(E) Each node in the tree has only one parent node.<br>【95 台大工工（丙），醫工（戊），電機（丙），電子（丙）】<br>ANS<br>(B)(C) |
| | 需修正為 |
| | 試題 3<br>Which of the following is (are) true about the “tree” data structure?<br>(A) Number of internal nodes = number of external (leaf) nodes – 1,<br>(B) Number of edges = number of nodes – 1,<br>(C) For each external (leaf) node, there exists only one trace from the root node to this external node,<br>(D) Each node in the tree can be a root node of a subtree,<br>(E) Each node in the tree has only one parent node.<br>【95 台大工工（丙），醫工（戊），電機（丙），電子（丙）】<br>ANS<br>(B)(C)(D) |

| P. 204 | 原錯誤處 |
|---|---|
|  | 試題 13<br>Draw a Binary tree for the following traversal, respectively.<br>(1) Postorder traversal = A D H E B F I C J G<br>Inorder traversal = A H D E G J B F C I (5%)<br>(2) Postorder traversal = E I B D G J C F H A<br>Inorder traversal = D G B J C A H F I E (5%) 【102 台北大學資管所資料結構】 |
|  | 需修正為 |
|  | 試題 13<br>Draw a Binary tree for the following traversal, respectively.<br>(1) Postorder traversal = A D H E B F I C J G<br>Inorder traversal = A H D E G J B F C I (5%)<br>(2) Preorder traversal = E I B D G J C F H A<br>Inorder traversal = D G B J C A H F I E (5%) 【102 台北大學資管所資料結構】 |

P. 208

## 原錯誤處

試題 20 解答：

ANS

(1) $2^{i-1}$

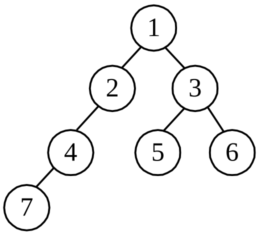
(2) $2^{k}-1$

(3) $n_0=n_2+1$

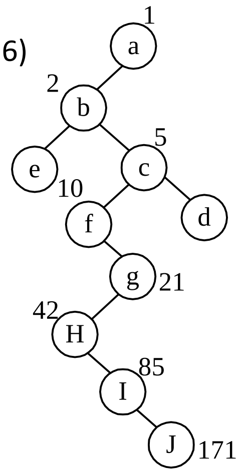
(4) $\lceil \log_2(n+1) \rceil$

(5) $n+1$

(9)

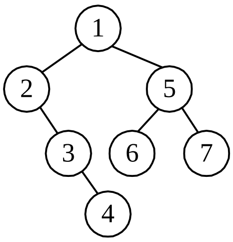


(6)



(7) $g=21$ ， $j=171$

(8)



## 需修正為

ANS

(1) $2^{i-1}$

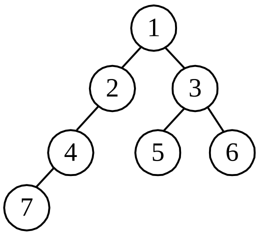
(2) $2^{k}-1$

(3) $n_0=n_2+1$

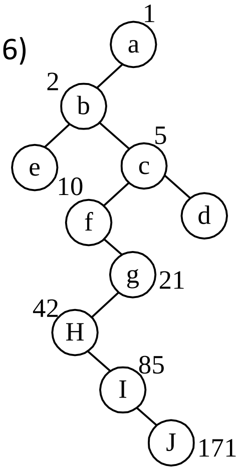
(4) $\lceil \log_2(n+1) \rceil$

(5) $n+1$

(9)

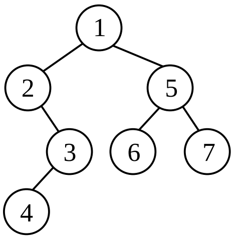


(6)



(7) $g=21$ ， $j=171$

(8)

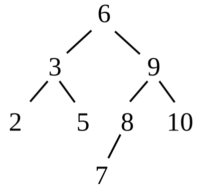


<table>
<tr><td>P. 220</td><td>原錯誤處</td></tr>
<tr><td></td><td>
試題 39 解答：

ANS

(1) 2, 7, 8, 3, 10, 9, 6

(2)



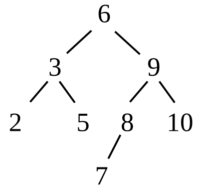
(3) Preorder (A[2 ∗ i]);

Preorder (A[2 ∗ i + 1]);
</td></tr>
<tr><td></td><td>需修正為</td></tr>
<tr><td></td><td>
試題 39 解答：

ANS

(1) 2,3,7, 8,10, 9, 6

(2)



(3) Preorder (A[2i]);

Preorder (A[2i + 1]);
</td></tr>
</table>

P. 227

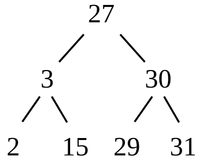
## 原錯誤處

試題 47 解答(3)：

(3) 從 H Delete Two keys 即 31 與 29

Then 插入 31，29 到 T 中

∴T:

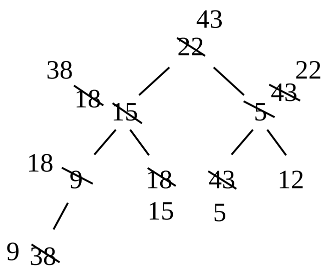


∴Array :

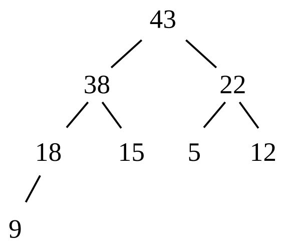
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 27 | 3 | 2 | 15 | 30 | 29 | 31 |

試題 48 解答(1)：

(1)



∴



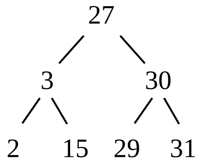
## 需修正為

試題 47 解答(3)：

(3) 從 H Delete Two keys 即 31 與 29

Then 插入 31，29 到 T 中

∴T:

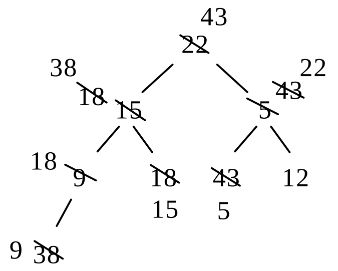


∴Array :

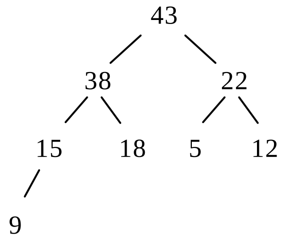
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 27 | 3 | 30 | 2 | 15 | 29 | 31 |

試題 48 解答(1)：

(1)



⊢



| P.228 | 原錯誤處 |
|---|---|
|  | 試題 50<br>What is a heap? How is a heap different from a binary search tree? Draw the resultant heap after you insert into the initially empty heap the following values, 10, 9, 3, 5, 2, 6. What is the resultant heap if the insertion order is changed to 6, 3, 5, 10, 9, 2? (12%)<br>【104 政大資科資料結構及演算法】 |
|  | 需修正為 |
|  | 試題 50<br>What is a heap? How is a heap different from a binary search tree? Draw the resultant heap after you insert into the initially empty heap the following values, 10, 9, 3, 5, 2, 6. What is the resultant heap if the insertion order is changed to 6, 3, 5, 10, 9, 27? (12%)<br>【104 政大資科資料結構及演算法】 |

P. 273

## 原錯誤處

試題 19 解答

ANS

理論：

A graph G (directed or not) contains a cycle if and only if a DFS of G yields a back edge.

```
dfsAll(G){
        for each vertex v of G{
                color[v] = “white”;
                parent[v] = null;
         }
        for each vertex v of G {
            if( color[v] == “white”)
                    if(DFS(v)) {Output(“Cycle detected”); exit;};
        }
        Output(“No cycle detected”);
}

boolean DFS(v){
        color[v] = “gray”;
        for( each vertex w adjacent to v ){
                    if (color[w] == “white”){
                            parent[w] = v;
                            DFS(w);
                       }
                else if( color[w] == “gray” and parent[w] !=   v)
                            return true;            // cycle detected
         }
        color[v] = black;
        return false;               // no cycle detected in this component
}
```

Time：因為最多拜訪 N 個頂點後就可以知道有無 cycle 存在，所以時間為 O(V)

## 需修正為

試題 19 解答

ANS

理論：

A graph G (directed or not) contains a cycle if and only if a DFS of G yields a back edge.

```
dfsAll(G){
        for each vertex v of G{
              color[v] = "white";
              parent[v] = null;
         }
        for each vertex v of G {
          if( color[v] == "white")
               if(DFS(v)) {Output("Cycle detected"); exit;};
        }
        Output("No cycle detected");
    }

    boolean DFS(v){
        color[v] = "gray";
        for( each vertex w adjacent to v ){
               if (color[w] == "white"){
                     parent[w] = v;
                     DFS(w);
                }
              else if( color[w] == "gray" and parent[v] !=   w)
                     return true;          // cycle detected
         }
        color[v] = black;
        return false;            // no cycle detected in this component
    }
```

Time：因為最多拜訪 N 個頂點後就可以知道有無 cycle 存在，所以時間為 O(V)

| P. 278 | 原錯誤處 |
|---|---|
| | 試題 23 解答<br>ANS<br>(1) practice by yourself<br>(2) 相鄰多元串列(practice)<br>(3) DFS(a)有向圖：adghfceb<br>(4) DFS(c)無向圖：cfhgdabe<br>(5) BFS(a)有向圖：adbeghfc<br>(6) BFS(c)無向圖：cfehbagd |
| | 需修正為 |
| | 試題 23 解答<br>ANS<br>(1) practice by yourself<br>(2) 相鄰多元串列(practice)<br>(3) DFS(a)有向圖：adghfceb<br>(4) DFS(c)無向圖：ceabdghf<br>(5) BFS(a)有向圖：aebdfhgc<br>(6) BFS(c)無向圖：cefabhdg |

| P.281 | 原錯誤處 |
|---|---|
|  | 試題27解答(2)<br>(2) Kruskal’s algo.<br>steps：①挑(B, R)<br>②挑(B, F)<br>③挑(F, P)<br>④挑(K, E)<br>⑤放棄(E, F) |
|  | 需修正為 |
|  | 試題27解答(2)<br>(2) Kruskal’s algo.<br>steps：①挑(B, R)<br>②挑(B, F)<br>③挑(F, P)<br>④挑(K, E)<br>⑤挑(E, F) |

P. 289

## 原錯誤處

試題 37 解答

ANS

(A)(C)

| iteration | S | U selected | Dist | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| init | | | | 0 | ∞ | ∞ | 7 | 15 | ∞ | 5 | ∞ |
| 1 | A | G | | 0 | ∞ | ∞ | 7 | 15 | ∞ | 5 | 18 |
| 2 | AG | D | | 0 | 10 | ∞ | 7 | 13 | ∞ | 5 | 18 |
| 3 | AGD | B | | 0 | 10 | 12 | 7 | 13 | ∞ | 5 | 18 |
| 4 | AGDB | C | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 18 |
| 5 | AGDBC | E | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 17 |
| 6 | AGDBCE | F | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 17 |

## 需修正為

試題 37 解答

ANS

(A)(B)(C)

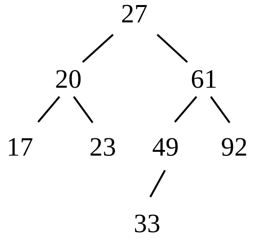
| iteration | S | U selected | Dist | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| init | | | | 0 | ∞ | ∞ | 7 | 15 | ∞ | 5 | ∞ |
| 1 | A | G | | 0 | ∞ | ∞ | 7 | 15 | ∞ | 5 | 18 |
| 2 | AG | D | | 0 | 10 | ∞ | 7 | 13 | ∞ | 5 | 18 |
| 3 | AGD | B | | 0 | 10 | 12 | 7 | 13 | ∞ | 5 | 18 |
| 4 | AGDB | C | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 18 |
| 5 | AGDBC | E | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 17 |
| 6 | AGDBCE | F | | 0 | 10 | 12 | 7 | 13 | 15 | 5 | 17 |

P. 456

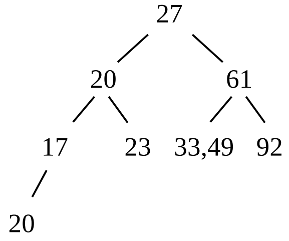
## 原錯誤處

試題 26

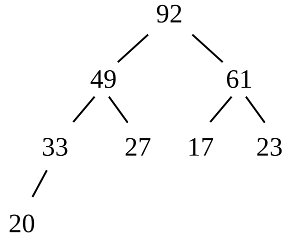
Insert a sequence of keys (27, 49, 17, 20, 61, 23, 92, 33) into a data structure that has no keys in the beginning. The results are depicted as follows. Please answer the corresponding data structure for (a), (b), and (c), respectively. (15%)



(a)



(b)



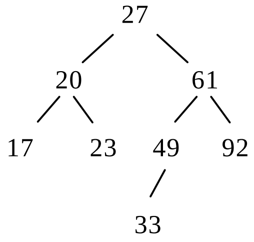
(c)

【100 成大資工程式設計】

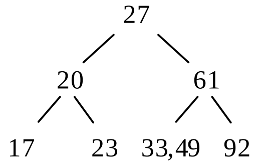
## 需修正為

試題 26

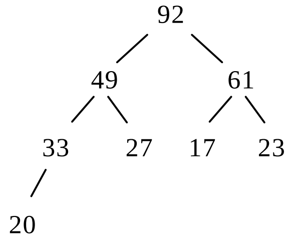
Insert a sequence of keys (27, 49, 17, 20, 61, 23, 92, 33) into a data structure that has no keys in the beginning. The results are depicted as follows. Please answer the corresponding data structure for (a), (b), and (c), respectively. (15%)



(a)

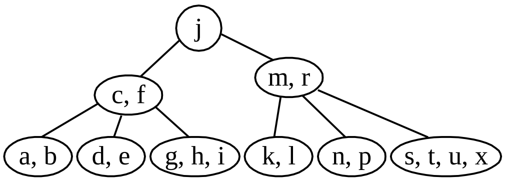
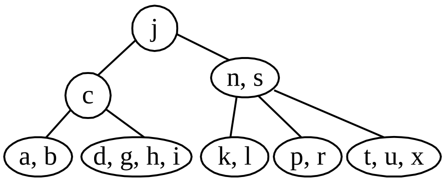
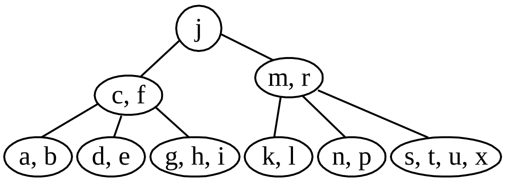
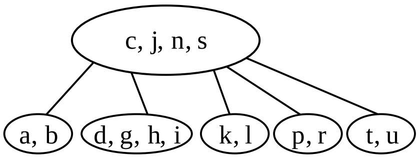


(b)



(c)

【100 成大資工程式設計】

| P. 460 | 原錯誤處 |
|---|---|
| | 試題 32 解答<br>(1)<br><br> j<br>c, f<br>m, r<br>a, b<br>d, e<br>g, h, i<br>k, l<br>n, p<br>s, t, u, x<br>(2) 假設 combine 與右兄弟 combine<br>刪 Non-leaf 時，用右子樹之最小值取代<br>則：<br><br> j<br>c<br>n, s<br>a, b<br>d, g, h, i<br>k, l<br>p, r<br>t, u, x |
| | 需修正為 |
| | 試題 32 解答<br>(1)<br><br> j<br>c, f<br>m, r<br>a, b<br>d, e<br>g, h, i<br>k, l<br>n, p<br>s, t, u, x<br>(2) 假設 combine 與右兄弟 combine<br>刪 Non-leaf 時，用右子樹之最小值取代<br>則：<br><br> |