

111 年公務人員高等考試三級考試試題

等 別：三等考試

類 科：資訊處理

科 目：資料結構

甲、申論題部分：

一、以下是一中序運算式 (Infix expression) 轉換 (Convert) 成後序運算式 (Postfix expression) 的演算法

```
operstk = the empty stack;
while(not end of input){
  symb = next input character;
  if(symb is an operand)
    add symb to the postfix string;
  else{
    while(!empty(operstk) && precedence(stacktop(operstk),symb)){
      topsymb = pop(operstk);
      add topsymb to the postfix string;
    } /*end while*/
    if (empty(operstk) || symb != '(')
      push(operstk, symb);
    else
      topsymb = pop(operstk);
    } /*end else*/
  } /*end while*/
while(!empty(operstk)){
  topsymb = pop(operstk);
  add topsymb to the postfix string;
} /*end while*/
```

其中資料結構：

“operstk”：用來儲存運算子的堆疊 (Stack)；

“stacktop(operstk)”：表示 top 指標所指堆疊 operstk 的運算子；

程序 (Procedures) 或函數 (Functions)；

“empty(operstk)”：檢查堆疊 operstk 是否為空的布林函數；

“pop(operstk)”：從堆疊 operstk 中取出一運算子；

“push(operstk, symb)”：將運算子 symb 存入堆疊 operstk；

“precedence(op1,op2)”：布林函數，定義在一沒有左右括弧的中序運算式中，op1 運算子出現在 op2 運算子的左邊時，當 op1 運算子優先順序不低於 op2 運算子，則設定成 TRUE，否則為 FALSE。例如，我們給定 precedence(*, +)=TRUE，precedence(+, +)=TRUE，precedence(+, *)=FALSE，為了處理運算式左右括弧，設定下列的 precedence:

precedence('(', op) = FALSE /*op 為任一運算子*/

precedence(op, '(') = FALSE /*op 為除')外的任一運算子*/

precedence(op, ')') = TRUE /*op 為除')外的任一運算子*/

precedence(')', op) = undefined /*op 為任一運算子*/

以中序運算式(2+3)*4 為例，執行上述演算法，依處理每一個運算子或運

算元時，輸出 postfix string 及 operstk 內容為何 (“eos”表示 end of string)？(25 分)

symbol	postfix string	operstk
((
2	2	(
+	2	+
3	23	(
)	23+	+
*	23+	(
4	23+4	+
eos	23+4*	empty

【解題關鍵】

1. 《考題難易》：★
2. 《破題關鍵》：本題為堆疊實作基本題，只要掌握利用堆疊將中序運算式轉換成後序運算式的演算法即可得到正確解答。
3. 《命中特區》：H1K16 資料結構講義 P103 完全命中。

【擬答】

我們可以利用一個堆疊來進行中置式與後置式的轉換，碰到運算元直接輸出，碰到運算子則比較其與堆疊頂端運算子的運算優先順序(precedence)，若 precedence 為 False，需將堆疊中運算子彈出來；否則就要把堆疊外的運算子壓入堆疊。

- (1)取出 e 中的下一個單元。
- (2)若取出單元為'Eos'則結束，只要將堆疊中剩餘元素依序取出即完成。
- (3)若取出單元為運算元則直接輸出。
- (4)若取出單元為')'，則將堆疊中元素依序輸出到'{'為止(但是必須將')'消去，因為在後置式中不需要括號來顯示優先順序與結合性)。
- (5)若取出單元為其他運算子，則比較其與堆疊頂端運算子的運算優先順序，若為 False，則需將堆疊中運算子取出來，直到 precedence 變成 True，再將堆疊外的運算子加入堆疊。
- (6)回到步驟(1)繼續。

operstk 中 top 指標指向堆疊最上元素

symbol	postfix string	operstk
((
2	2	(
+	2	+
3	23	(
)	23+	+
*	23+	(
4	23+4	+
Eos	23+4*	empty

112年 虛實整合

多元學習新型態

突破傳統上課形式 **5大方式** 彈性又便利

| 面授學習 | 直播學習 | 在家學習 | 視訊學習 | Wifi學習 |

◆學習◆
零時差

同類科各班別
皆可同步直播上課

◆服務◆
零死角

服務緊貼需求
隨時掌握學習狀況

線上
課業諮詢



老師
申論批閱



雙師資
雙循環



多元
補課方式



上榜生
經驗親授



時事
專題講座



歷屆試題
練習



班導師
制度



各班服務略有不同，詳情請洽全國志光、保成、學儒門市

二、利用鏈結串列 (Linked list) 實做佇列 (Queues)，給予如下鏈結串列節點及佇列定義，front 指標指在串列第一個節點，rear 指標指在串列最後一個節點，請使用 C 語言完成 insert(pq, x) 程序，將整數值 x 加入 (Insert) 到佇列，程式需檢查佇列加入前是否為空的鏈結串列，可使用函數 getnode() 配置 (Allocate) 一新節點。(25 分)

```

struct node{
    int info;
    struct node *next;
};
typedef struct node *NODEPTR;

struct queue{
    NODEPTR front, rear;
};
struct queue q;

NODEPTR getnode()
{
    NODEPTR p;
    p = (NODEPTR)malloc(sizeof(struct node));
    return(p);
}

insert(pq, x)
struct queue *pq;
int x;
{
    NODEPTR p;
}
    
```

- 【解題關鍵】**
1. 《考題難易》：★★★★
 2. 《破題關鍵》：本題為鏈結串列基本實作題，只要掌握鏈結串列實做佇列方法即可得到正確解答。
 3. 《命中特區》：H1K16 資料結構講義 P192-193 完全命中。

【擬答】

```

insert(pq, x)
struct queue *pq;
int x;
{
  NODEPTR p;
  p=getnode();
  p->info = x;
  p->next=NULL;
  if(pq->front==NULL)
    pq->front = p;
  else
    pq->rear->next=p;
  pq->rear=p;
}

```

忘光學儒保成

到底怎樣才能 **輕鬆考取?**

快來掌握 **8** 大課程密招

<p>法科架構班 結合實務例子 建構法科概念</p>	<p>扎實正規班 完整堂數 循序漸進</p>	<p>工科全科班 公職+國營 一次到位</p>	<p>作文實戰班 強化寫作架構 理清邏輯概念</p>
<p>主題題庫班 主題教學 考點分析</p>	<p>精華總複習 掌握考點 增強實力</p>	<p>全真模擬考 比照真實考試 檢視應考實力</p>	<p>考前關懷講座 名師最終提點 觀念更加清晰</p>

7/29(五)前, 憑111年高普考准考證報名課程享考生折扣

三、一個二元搜尋樹 (Binary search tree) 的前序追蹤 (Preorder traversal) 結果如下：14, 4, 3, 9, 7, 5, 15, 18, 16, 17, 20 請建構此二元搜尋樹。接著利用如下 C 語言對二元樹節點的宣告，使用 C 語言寫一遞迴程式 sortTree (NODEPTR tree)，輸入二元樹的根節點，來處理此二元樹的節點資料，並將資料依由小至大輸出。(25 分)

```

struct node{
    int info;
    struct node *left;
    struct node *right;
}
typedef struct node *NODEPTR;

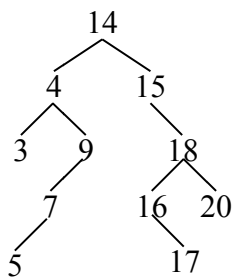
void sortTree(NODEPTR tree){
}
    
```

【解題關鍵】

1. 《考題難易》：★★
2. 《破題關鍵》：本題為二元搜尋樹概念題，只要掌握二元搜尋樹中序走訪可得由小到大排序概念，搭配反推二元樹手法即可得到正確解答。
3. 《命中特區》：H1K16 資料結構講義 P243 (反推二元樹)、P254 (BST) 完全命中。

【擬答】

(一)此二元搜尋樹之前序追蹤為 14,4,3,9,7,5,15,18,16,17,20；中序追蹤結果即為排序結果，故為 3,4,5,7,9,14,15,16,17,18,20。因此可以反推二元搜尋樹為



(二)二元搜尋樹資料依序由小至大輸出，只要使用中序走訪即可達成，故可用下面的遞迴程序：

```

struct node{
    int info;
    struct node *left;
    struct node *right;
}
typedef struct node *NODEPTR;

void sortTree(NODEPTR tree) {
    if (tree != NULL) {
        sortTree(tree->left);
        printf("%d ", tree->info);
        sortTree(tree->right);
    }
}
    
```


公職王歷屆試題 (111 高考三等)

四、用 $G=(V,E)$ 表示一個無方向性圖形，其中 V 是點的集合， E 是一組節點 (Vertices) 形成邊及對應權重 (Weights) 所組成的集合。今有一圖形 $G=(V,E)$ ， $V=\{0,1,2,3,4,5\}$ ，圖形的邊與權重值以如下的定義儲存對應連接矩陣 (Adjacency matrix) 表示中的值

```
#define MAX_EDGES 100
typedef struct {
    int col;
    int row;
    int weight;
} edge;
edge a[MAX_EDGES];
```

已知陣列 a 儲存對應連接矩陣相連接邊的內容如下： $a = \{(3, 0, 2), (4, 0, 1), (5, 0, 20), (2, 1, 7), (5, 1, 24), (3, 2, 15), (4, 2, 10), (5, 2, 25), (4, 3, 3)\}$ 。請畫出陣列 a 所儲存的圖形，然後，利用 Prim 演算法從節點 0 開始依加入其它節點的順序，畫出此圖之最小擴張樹 (Minimumspanning tree)，並計算其最低權重或成本值。(25 分)

【解題關鍵】

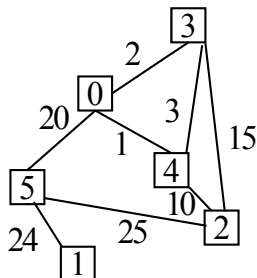
1. 《考題難易》：★★
2. 《破題關鍵》：本題為圖形基本操作題，只要了解以 Prim 演算法求最小擴張樹即可得到正確解答。
3. 《命中特區》：H1K16 資料結構講義 P352-353 完全命中。

【擬答】

(一)依照所給資料，此連接矩陣為

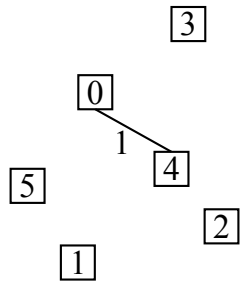
	0	1	2	3	4	5
0	$+\infty$	$+\infty$	$+\infty$	2	1	20
1	$+\infty$	$+\infty$	7	$+\infty$	$+\infty$	24
2	$+\infty$	7	$+\infty$	15	10	25
3	2	$+\infty$	15	$+\infty$	3	$+\infty$
4	1	$+\infty$	10	3	$+\infty$	$+\infty$
5	20	24	25	$+\infty$	$+\infty$	$+\infty$

故陣列 a 儲存的圖形為

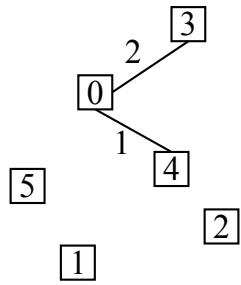


(二)根據 Prim 演算法從節點 0 開始的順序

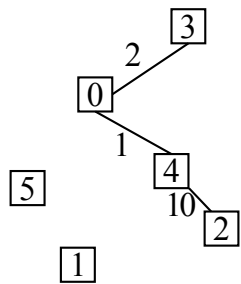
1. 加入(4,0,1)



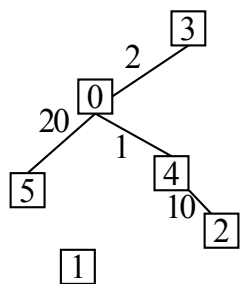
2. 加入(3,0,2)



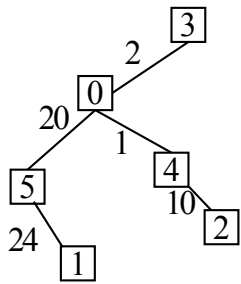
3. 加入(2,4,10)



4. 加入(5,0,20)



5. 加入(5,1,24)



完成最小成本擴張樹，總成本為 $1+2+10+20+24=57$ 。

公
職
王