

《資料結構》

試題評析	<p>第一題：測驗合併排序所使用的兩種選擇樹。考前若有準，取得分數不難。</p> <p>第二題：關節點的相關計算與判定方法，屬於以往較少考的內容，若有複習而取分，相對獲得更多優勢。</p> <p>第三題：稀疏矩陣的表示法，以及其快速轉置的做法，有詳細準備的考生應可取得分數。</p> <p>第四題：樹的處理與相關計算，在本試題中，屬於較簡單的題目。</p> <p>第五題：較少考的內容，而且需要逐步推導，亦屬於較繁複的題目，若考前確實準備而拿到分數，可比其他考生得到相當多的競爭優勢。</p> <p>綜觀此份試題涵蓋範圍十分廣泛，反應出近年國家考試的命題趨勢，未來考生需做十分周全的準備，方能在資料結構一科，得到應有的分數。</p>
考點命中	<p>第一題：《高點資料結構》，王致強編撰，頁 9-45~9-49。</p> <p>第二題：《高點資料結構》，王致強編撰，頁 8-52~8-56。</p> <p>第三題：《高點資料結構》，王致強編撰，頁 3-64~3-66。</p> <p>第四題：《高點資料結構》，王致強編撰，頁 6-27 精選例題 32、頁 6-33 精選例題 38。</p> <p>第五題：《高點資料結構》，王致強編撰，頁 11-59~11-65。</p>

一、(一)假設有 8 個排序好的數列(如圖 1)，請建構一 loser tree 並顯示取出前 4 個最小值之 loser tree 變化。(16 分)

(二)請說明 loser tree 和 winner tree 差異為何？(4 分)

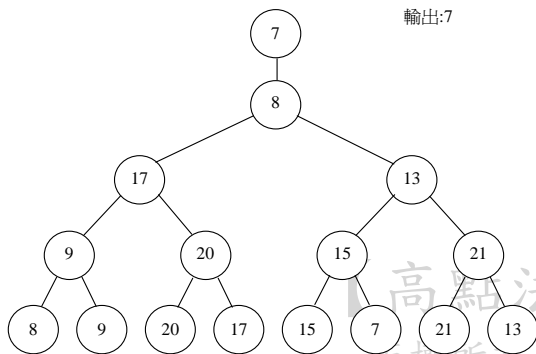
8	9	20	17	15	7	21	13
23	12	70	25	45	10	31	32
27	38	80	28	48	19	36	33

圖 1

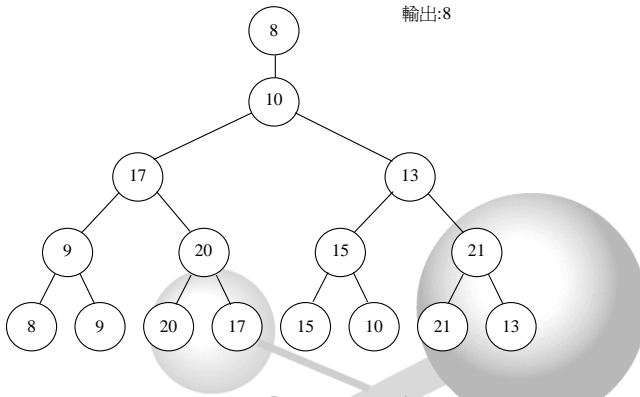
【擬答】

(一)

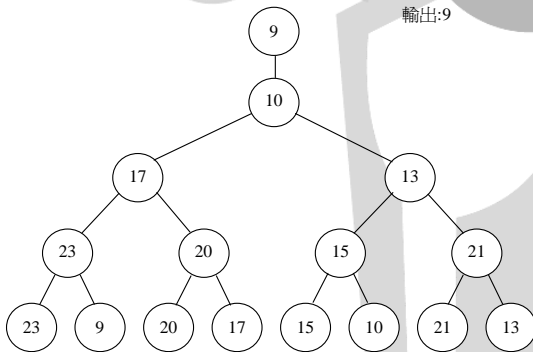
(1)



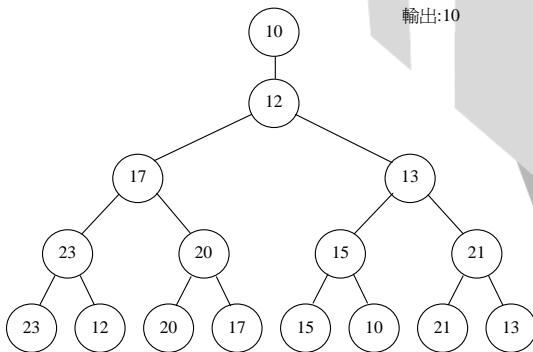
(2)



(3)



(4)



- (二) loser' s tree 記錄兩個子節點較大者；而較小者住其父節點移動。
winner' s tree 只記錄兩個子節點較小者；較大者仍停留在子節點。
處理方式如下：

loser' s tree：

每次輸出最小的資料之後，loser' s tree 由最小的資料所在的 run 遞補下一個資料，往 root 的路徑往上一路比較，每經過一個節點就與該節點原有的資料比較，將較大的留在該節點上；而較小的繼續往 root 方向移動，此程序一直到 root 為止，最後最小的資料在 root 上方的節點。

winner' s tree：

每次輸出最小的資料之後，winner' s tree 也是由最小的資料所在的 run 遞補下一個資料至終端節點，然後與其 sibling 比較，將較小的資料移至父節點，然後上移的資料繼續與 sibling 比較，依此類推，一直到 root 為止，最後 root 即為最小的資料。

二、(一)請利用 dfn (depth-first number) 及 low (the lowest depth-first number) 值，找出圖 2 所有之關節點 (articulation points)。假設利用深度優先搜尋法 (depth first search) 讀取節點之順序為 4-2-1-3-5-6-8-9-7，也就是節點 4 之 dfn 值為 1，節點 2 之 dfn 值為 2，節點 1 之 dfn 值為 3，依此類推。(15 分)

(二)請說明如何判斷那一節點為關節點？ low 計算之公式為何？(5 分)

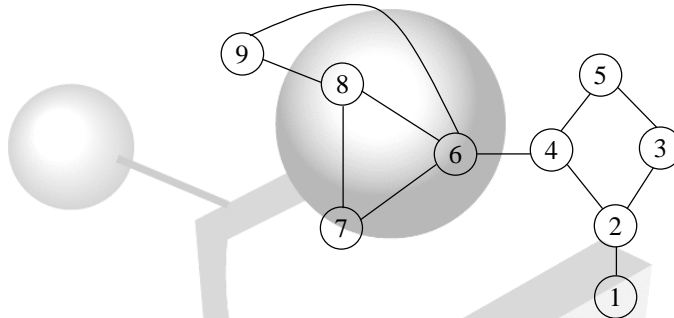
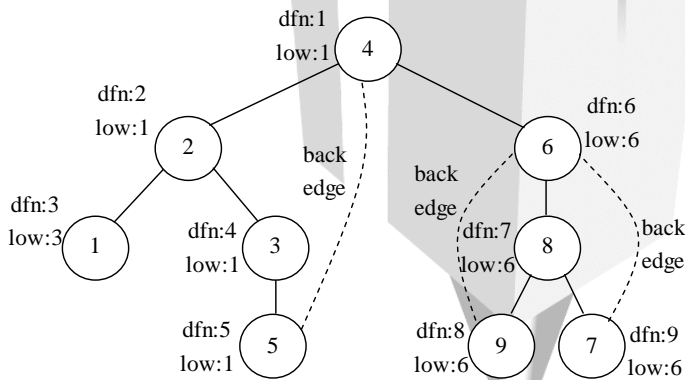


圖 2

【擬答】

(一)dfs spanning tree 如下:



dfn 與 low 值如下表所示：

vertex	1	2	3	4	5	6	7	8	9
dfn	3	2	4	1	5	6	9	7	8
low	3	1	1	1	1	6	6	6	6

articulation points 有 2,4,6 三個。

(二)判斷 articulation point 的規則，是由 dfs spanning tree 中，即據頂點的 dfn 與 low 值來判斷，分為兩種情況，分別使用不同規則：

- 1.如果 vertex u 是 root：只要有超過一棵以上的 sub-trees，則 root 即為 articulation point，如本題的頂點 4。
- 2.如果 vertex u 不是 root：在 dfs spanning tree 中，如果 u 有任一個子節點 v 的 $low[v] \geq dfn[u]$ ，則頂點 u 即為 articulation point，如本題的頂點 2 與頂點 6。

三、給一稀疏矩陣 (sparse matrix) M 如圖 3 所示。

(一)請以 3-tuple form ($i, j, value$) 來表示此矩陣 M 。(6 分)

(二)針對(一)之 3-tuple form，請設計一有效率而時間複雜度不大於 $O(columns + terms)$ 之快速矩陣轉置 (fast matrix transposing) 演算法。其中 $columns$ 為欄的數目， $terms$ 為非零項目的數目。以圖 3 所示， $columns = 4$ 、 $terms = 6$ 。(14 分)

5	0	0	11
0	41	0	0
0	0	0	0
63	0	23	0
0	0	0	12

圖 3

【擬答】

(一)

	i	j	value
0	5	4	6
1	1	1	5
2	1	4	11
3	2	2	41
4	4	1	63
5	4	3	23
6	5	4	12

(二)使用下面的 fast transpose 方法：

1. $b[0,i]=a[0,j]$; $b[0,j]=a[0,i]$; $b[0,value]=a[0,value]$;
2. if $a[0,value]>0$ then begin
3. for $col:=1$ to $a[0,j]$ do $RowSize[col]:=0$;
4. for $t:=1$ to $a[0,value]$ do $RowSize[a[t,j]]:=RowSize[a[t,j]]+1$;
5. $RowStart[1]:=1$;
6. for $col:=2$ to $a[0,j]$ do $RowStart[col]:=RowStart[col-1]+RowSize[col-1]$;
7. for $t:=1$ to $a[0,value]$ do begin
8. $b[RowStart[a[t,j]],i]:=a[t,j]$;
9. $b[RowStart[a[t,j]],j]:=a[t,i]$;
10. $b[RowStart[a[t,j]],value]:=a[t,value]$;
11. $RowStart[a[t,j]]:=RowStart[a[t,j]]+1$;
12. end
13. end;

時間分析：

line 3: $O(columns)$

line 4: $O(terms)$

line 6: $O(columns)$

line 7-12: $O(terms)$

總時間: $O(columns+terms)$ 。

【高點法律專班】

四、(一)請設計一演算法，將一個二元樹 (binary tree) 每一節點之左子樹和右子樹對調 (swap)，如下圖 4 所示。(8 分)

(二)假設一 n 個 nodes 之 k -ary tree (即分支度為 k 之樹) T ，每一個 node 有一固定大小之欄位如下，請說明共有多少欄位是 Null？(8 分)

資料	欄位 1	欄位 2	...	欄位 k
----	------	------	-----	--------

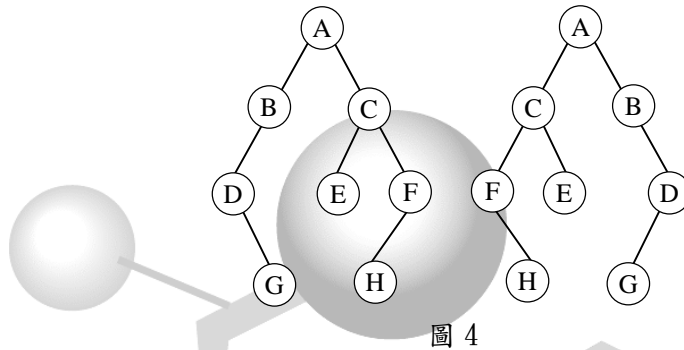


圖 4

【擬答】

- (一)
- ```

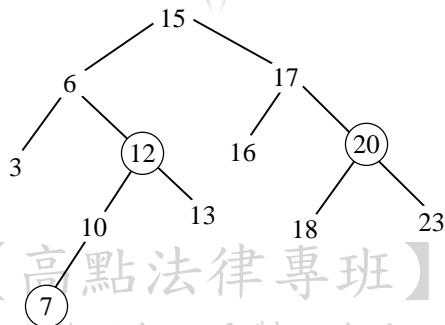
1. void swaptree(treeptr t)
2. { treeptr p;
3. if (t!=NULL)
4. {
5. p=t->left; t->left=t->right; t->right=p;
6. swaptree(t->left);
7. swaptree(t->right);
8. }
9. }

```
- (二)整棵樹共有  $n \times k$  個指標欄位，有被用來指向子節點的有  $n-1$  個指標欄位，因此空指標欄位(即未用到的指標)為  $n \times k - (n-1) = n \times (k-1) + 1$  個。

五、(一)請說明紅黑樹 (red-black tree) 之特性。(4 分)

(二)建立一紅黑樹，其數字依序為 10、72、14、68、20、58、30、50、65、63。(10 分)

(三)請一步一步刪除圖 5 紅黑樹之節點，依序為 10、18、3、16、13、12、17。其中在圖 5 之節點 7、12、20 為紅色節點。(10 分)



【高點法律專班】  
版權所有，重製必究！

圖 5

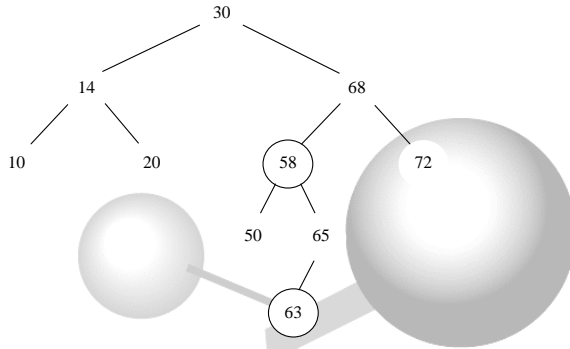
## 【擬答】

- (一) red-black tree 是一棵 binary search tree，且必須滿足下面特性：
- (1)每個節點可以是 red node 就是 black node。
  - (2)root 一定是 black node。
  - (3)所有的 leaves(external node)皆為 black nodes。

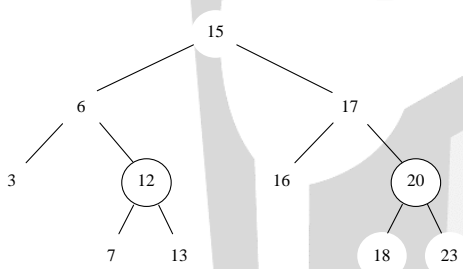
(4) 一個 red node，其 children 皆為 black node。

(5) 每個節點到其所有後代 leaves 的每一條路徑上，皆包含相同數目的 black node。

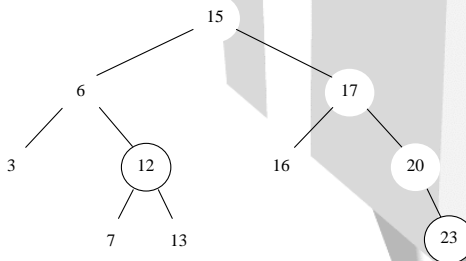
(二)



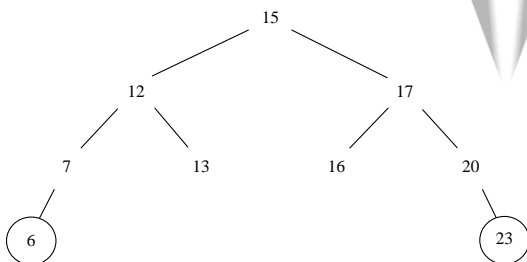
(三) 刪除 10



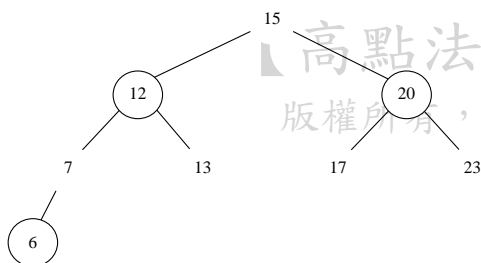
刪除 18



刪除 3

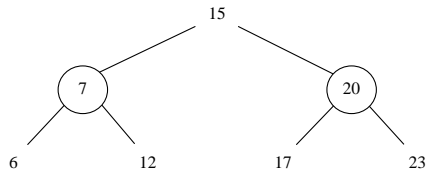


刪除 16

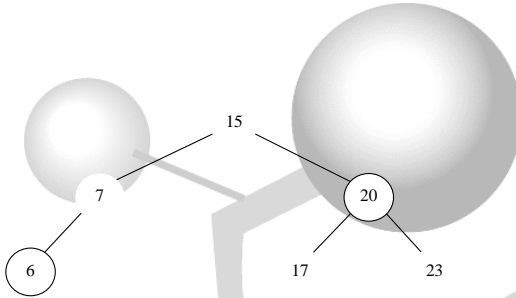


【高點法律專班】  
版權所有，重製必究！

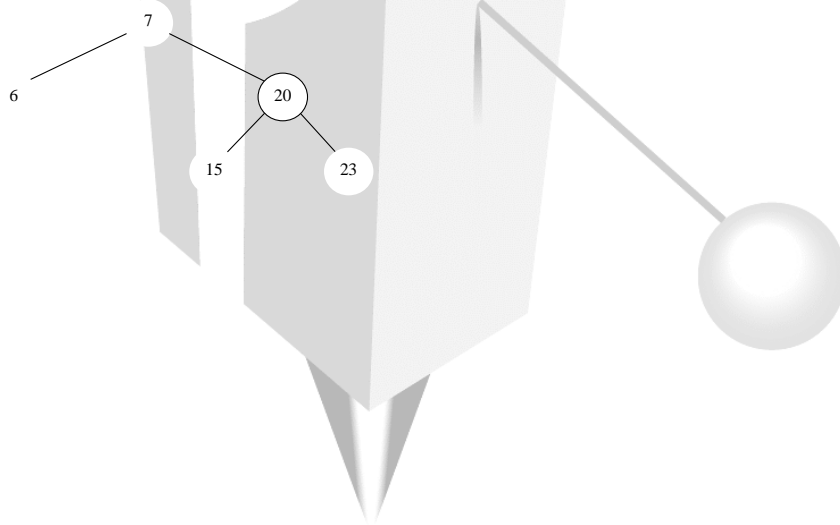
刪除 13



刪除 12



刪除 17



【高點法律專班】

版權所有，重製必究！