

《資料結構》

試題評析

今年高考資訊人員資料結構考題，十分平易，應考者應可拿到不錯的分數，但也較無法測驗出考生的程度。

第一題是基本遞迴程式設計的問題，為最大公因式的簡單變形，對大部分考生來說均不困難。第二題則是堆疊、佇列與雙向佇列的排列問題，雖是基本的問題，但排列組數較多，將考驗考生的熟悉度，堆疊須利用快速檢驗法解答，雙向佇列則應利用「所有堆疊可以排列的結果，雙向佇列亦可排列出」之方法，來減少所需檢查的組數，才能節省時間。第三題為各種樹的節點個數計算，可以輕易拿分，但要注意第(二)小題的條件，與平常記憶的公式稍有出入，要小心計算。第四題是二元樹追蹤的分析，亦容易取分。第五題是二元堆積的基本觀念題，拿分也不難。預測今年應考者資料結構一科的分數應普遍提高，拿到 85 分以上應是很平常的分數。

一、數學上求兩數的最大公因數(Greatest Common Divisor, 簡稱GCD)可使用歐幾里德(Euclid)的輾轉相除法來完成。規則是「兩數 m 與 n 的最大公因數等於這兩數的差和較小數的最大公因數」，由此可看出遞迴規則。請寫一個遞迴程式或演算法來計算 m 與 n 兩數($m > n$)的最大公因數。(20分)

答：使用 C 語言來撰寫如下：

```
int gcd(int m, int n)
{
    if (n==0) return m;
    else if (m>=n) return gcd(m-n, n);
    else return gcd(n-m, m);
}
```

二、試說明將 1, 2, 3, 4 四個數字依此一次序分別經由堆疊(stack)、佇列(queue)、與雙向佇列(deque)做排列，問各有多少種不同的排列方法？請也分別寫出這些排列的結果。(20分)

答：(一)以 stack 排列共有 14 組：1234、1243、1324、1342、1432、2134、2143、2314、2341、2431、3214、3241、3421、4321。

(二)以 queue 排列只有 1 組：1234。

(三)以 deque 排列全部 $4! = 24$ 組皆可以排列出來：1234、1243、1324、1342、1423、1432、2134、2143、2314、2341、2413、2431、3124、3142、3214、3241、3412、3421、4123、4132、4213、4231、4312、4321。

三、(一)一棵degree為 k 的二項式樹(binomial tree)有多少個節點(nodes)？(5分)

(二)一棵order為 m 的B-樹(b-tree)，失敗節點(failure nodes)落在第 n 層(level)，則這棵B-樹最少可存放多少個鍵(keys)？(5分)

(三)一棵degree為 m ，高度為 h 的查詢樹(search tree)最多有多少個節點？(5分)

(四)一棵深度(depth)為 h 的完整二元樹(complete binary tree)最少有多少個節點？(5分)

答：(一)degree 為 k 的二項式樹有 2^k 個節點。

(二)至少可以存放 $1 + 2(\lceil \frac{m}{2} \rceil - 1) + 2\lceil \frac{m}{2} \rceil(\lceil \frac{m}{2} \rceil - 1) + \dots + 2\lceil \frac{m}{2} \rceil^{n-3}(\lceil \frac{m}{2} \rceil - 1) = 2\lceil \frac{m}{2} \rceil^{n-2} - 1$ 個keys。

(三)最多有 $1 + m + m^2 + \dots + m^{h-1} = \frac{m^h - 1}{m - 1}$ 個節點。

(四)最少有 $1 + 2 + 2^2 + \dots + 2^{h-2} + 1 = 2^{h-1}$ 個節點。

四、找出在下述追蹤(traversal)皆具相同順序值之所有二元樹：

- (一)preorder及inorder。(7分)
- (二)preorder及postorder。(7分)
- (三)inorder及postorder。(6分)

答：(一)空的二元樹(empty binary tree) 或 右歪斜二元樹(right-skewed binary trees)。
(二)空的二元樹(empty binary tree) 或 單一節點的二元樹(single-node binary trees)。
(三)空的二元樹(empty binary tree) 或 左歪斜二元樹(left-skewed binary trees)。

五、(一)請詳細說明「二元堆積」(binary heap)此種資料型態支援那幾種運算？若以「陣列」(array)來「製作」(implement)二元堆積，這些運算在時間上的複雜度各為何？

(二)二元堆積裡所存放的元素裡，其最小元素只會出現在該二元堆積的那些地方？(20分)

答：(一)二元堆積(Max-heaps)支援二種運算：插入(Insertion) 與 刪除最大值>Delete the Largest element)。若用陣列來製作，此二種運算的時間複雜度皆為 $O(\log n)$ 。

(二)二元堆積(Max-heaps)裡的最小元素只會出現在葉節點(leaf node)上。

註：若是最小堆積(Min-heaps)則最小元素出現在樹根上。