

【資訊處理】

《資料結構》

試題評析

本次高考資訊人員資料結構試題相當平易，考生應可拿到不錯的分數。五個題目分別屬於雜湊法、搜尋法、B-tree、排序法及鏈結串列操作，沒有特別難的考題，較關鍵的題目應是第一題的第(三)小題，題目中提到 quadratic probing，根據再雜湊函數應該是 quadratic residue probing，宜小心作答。另外，第四題第(二)小題，必須了解快速排序避免最壞情形的方法，才能答對。綜觀今年試題，預估考生拿到80分以上應屬正常，程度較佳甚至可以拿到九十分以上的好成績。

一、假設輸入的資料是：7341, 3123, 1673, 4919, 4304, 9179, 1369，使用的雜湊函數 (hash function) 是 $f(x)=x \bmod 10$ ， x 是輸入的資料，而雜湊表格 (hash table) 的大小有10個位置，編號從0到9，每一位置只能儲存一筆資料。請分別回答下列的問題：

(一) 上述資料經由雜湊函數後，寫出雜湊表格的內容 (含溢位資料)。(4分)

(二) 當溢位處理方法 (overflow handling method) 使用線性探測 (linear probing) 時，請寫出雜湊表格的內容。(4分)

(三) 當溢位處理方法使用平方探測 (quadratic probing) 時，計算過程為 $f(x) \pm i^2 \bmod 10$ ，請寫出雜湊表格的內容。(5分)

(四) 當溢位處理方法使用雙重雜湊 (double hashing) 時，第二個雜湊函數是 $g(x)=7-(x \bmod 7)$ ，請寫出雜湊表格的內容。(7分)

答：

(一)

	0	1	2	3	4	5	6	7	8	9
表格		7341		3123	4304					4919
溢位				1673					9179	1369

(二)

	0	1	2	3	4	5	6	7	8	9
表格	9197	7341	1369	3123	1673	4304				4919

(三)

	0	1	2	3	4	5	6	7	8	9
表格	9179	7341		3123	1673	4304			1369	4919

(四)

	0	1	2	3	4	5	6	7	8	9
表格	1673	7341	9179	3123	4304	1369				4919

二、本題是討論循序找尋 (sequential search) 方法。假設陣列一共有 n 個元素，而循序找尋的演算法如下所示：

ALGORITHM Sequential Search ($A[0..n-1]$, key)

$i \leftarrow 0$

while ($i < n$ and $A[i] \neq \text{key}$) do

$i \leftarrow i + 1$

end

```

if i < n return i
else return - 1

```

(一)請寫出成功找尋的平均比較次數是多少？(5分)

(二)已知成功找尋的機率是 p ，請寫出成功與失敗的平均找尋次數是多少？(7分)

(三)若已知陣列中每一個元素的被讀取次數，例如：第一個元素被讀取5次，第二個元素被讀取12次，餘類推。請問如何可以減少成功找尋的平均比較次數？(8分)

答：

$$(一) \frac{1+2+3+\dots+n}{n} = \frac{n+1}{2}$$

$$(二) p \times \frac{n+1}{2} + q \times n = \frac{pn + p + 2qn}{2} = \frac{n + qn + p}{2} = \frac{n(1+q)}{2} + \frac{p}{2}$$

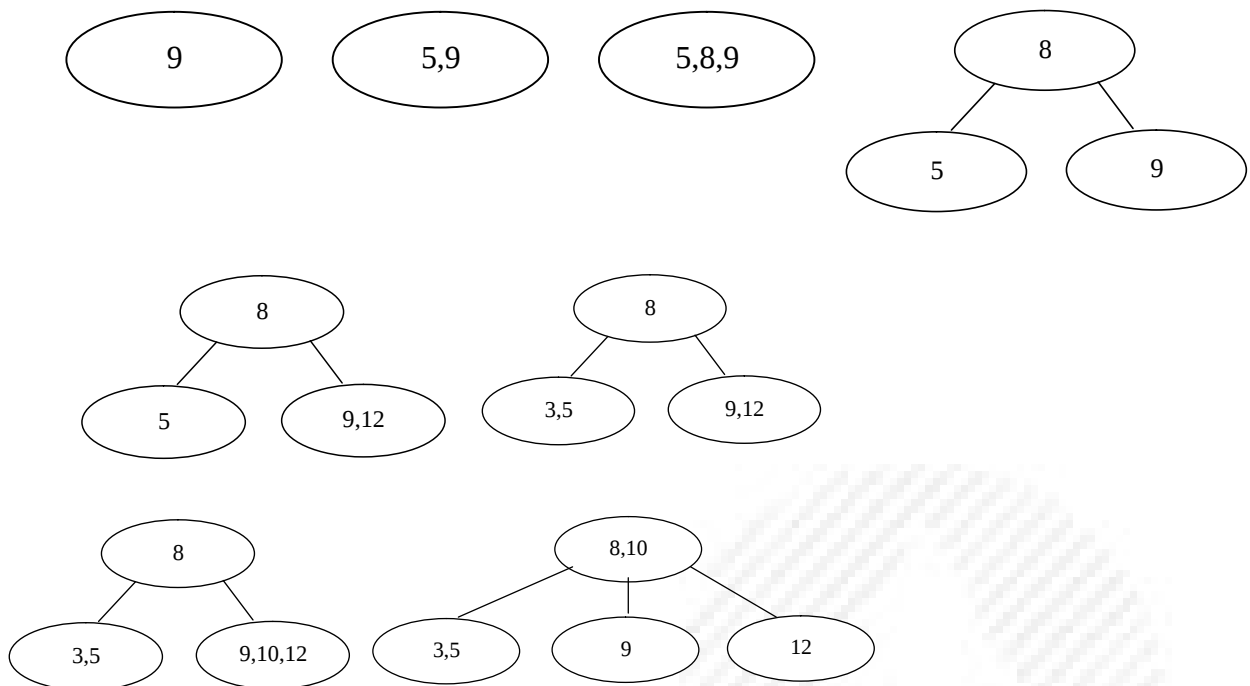
(三)將讀取次數較多的資料置於陣列開頭，依讀取次數由多到少排列置於陣列中，可以使平均比較次數有效地降低。

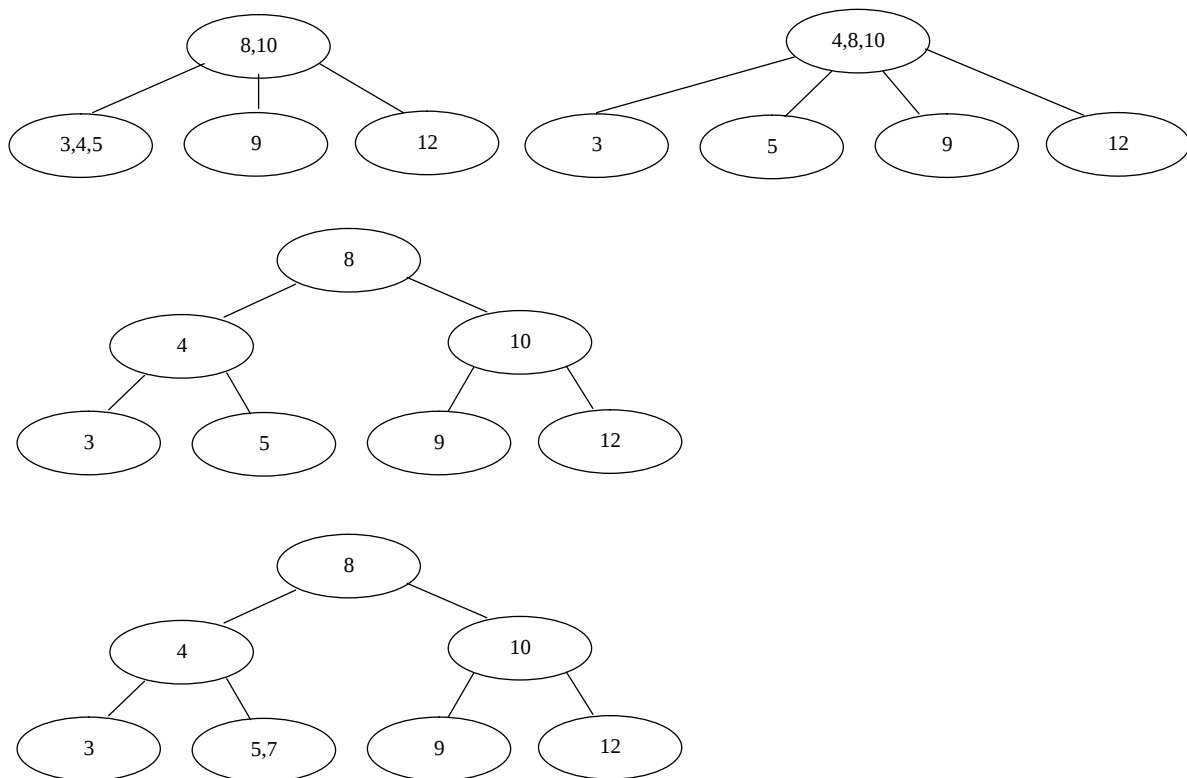
三、(一)給予一個串列資料：9, 5, 8, 12, 3, 10, 4, 7。請依序建造出2-3樹(2-3 tree)，並寫出建造的過程。(10分)

(二)若規定2-3樹的高度(height)是從樹根(root)到樹葉(leaf)的最長路徑。請寫出一個高度為 h 的2-3樹，能夠儲存的最多資料數目是多少？能夠儲存的最少資料數目是多少？(10分)

答：

(一)





(二)最多 $3^{h+1} - 1$ ；最少 $2^{h+1} - 1$ 。

四、快速排序法 (Quick sort) 是利用分割 (Partitioning) 技術，以遞迴方式做排序的一種高等排序方法。請回答下列的問題。

(一)說明分割技術的一般做法。(10分)

(二)快速排序法的最壞情況 (worst case) 需要 $O(n^2)$ 的時間，有一種改進方法可避免發生最壞情況，請說明這個改進做法。(10分)

答：

(一)分割時，取第一個陣列元素做基準值(pivot)，另外使用兩個整數變數*i*與*j*做指標，*i*由第二個元素往陣列尾端方向，找尋大於 pivot 的元素；而 *j* 則由最後一個元素往回頭方向找尋小於 pivot 的元素。

i 找到大於 pivot 的元素，且 *j* 找到小於 pivot 的元素後，將兩個元素交換，然後繼續同一程序，直到 *i* 移到 *j* 的右側時為止。

最後，再將 pivot 與第 *j* 個位置的資料交換，完成一次的 partition。

(二)可以用 median of three 的方法來選取基準值，也就是取第一個、最後一個與中間項，三者的中位數來當做基準值(pivot)，如此可以避免發生 worst case。

五、一個鏈結串列使用C語言宣告如下：

```
typedef struct node{
    int data;
    struct node *next;
}NODE;
```

```
NODE *new, *back, *pointer, *forward;
```

假設現在已經產生一個名叫new的鏈結串列共有n個節點，已知指標back是指向串列中間的某一個節點，而指標pointer是指向back的下一個節點。在下列的問題中指令(一)~(五)分別為何？（每小題4分，共20分）

問題一：刪除pointer所指向的節點。

```
_____ (一) _____=pointer->next;
free (pointer);
```

問題二：欲將pointer所指向節點的指標做反轉（reverse），指向前一個節點，此迴路（loop）可逐漸完成整個串列的反轉。

```
while (pointer->next != NULL){
    forward=_____ (二) _____;
    pointer->next =_____ (三) _____;
    _____ (四) _____= pointer;
    pointer=_____ (五) _____;
}
```

答：

- (一) back->next
- (二) pointer->next
- (三) back
- (四) back
- (五) forward