題型6–23 雜湊函數（hash function）

【重點說明】

一、雜湊函數

一般雜湊函數選擇具⑴計算簡單；⑵碰撞發生頻率低；⑶叢集（cluster）現象少等優點的函數來作雜湊函數。下面介紹最常用的雜湊函數：

➠ 除法（Division）：

這是一種常見的雜湊函數，就是將識別字x除以某個數值M，然後取其餘數做雜湊表中的位址，也就是利用模數（modulus）%的運算。求算的公式如下：$f_D(x) = x \% M$。

除法（Division）雜湊函數是一種常見的雜湊函數，就是將識別字x除以某個數值M，然後取其餘數做雜湊表中的位址，也就是利用模數（modulus）%的運算。求算的公式如下：$f_D(x) = x \% M$。在選擇數值M時，要注意儘可能不要讓識別字x的模數（modulus）結果相同，增加差異性，才能發揮雜湊函數的效果。

二、溢位（overflow）處理

透過雜湊函數求識別字位置時，若該桶已滿，則將產生溢位。當有溢位情況發生時，我們必須找出一個新的計算方法，稱為再雜湊函數（rehash function），整個處理過程稱為溢位處理。

1. 線性探測（linear probing）：

線性探測又稱為開放位址法（open addressing），是最常用的方法。原理為一旦發生碰撞，就往下一位置探測，如果下一位置仍然被占用，則繼續往下搜尋，直到找空白的位置為止。

➠ 線性探測的資料搜尋：

以線性探測的方式存放資料，搜尋資料時會發生三種可能的情形如下：

①經雜湊函數計算位置後，資料值與鍵值相同，表示搜尋成功。

②資料值與鍵值不相同，所以繼續往下探測，直到搜尋成功為止。

③在搜尋過程中遇到空白位置，這就表示搜尋失敗。

我們發現一個有趣的現象，就是當資料值若集中在某一區段，則每次該區段資料值插入時，發生碰撞的頻率將會快速增加，這種現象稱之為叢集（cluster）。

2. 再雜湊法（rehashing）：

設計一系列的雜湊函數f_1, f_2,…，當使用f_1發生溢位時，則改用f_2，若再發生溢位則改用f_3，依此類推，直到下一個雜湊函數不發生溢位為止。

三、雜湊法的優點

雜湊法的優點如下：

1. 資料不必先排序過。

2. 在沒有碰撞（collision）及溢位（overflow）的情形下，只需一次讀取即可，且其搜尋時間與資料量的多寡無關。

3. 具保密性，要知道雜湊函數後才能擷取資料。

4. 可做資料壓縮，利用適當的雜湊函數，可將資料壓縮到一個較小的範圍內，節省空間。

四、搜尋法綜合比較

常見搜尋法的比較表：

	循序搜尋法	二分搜尋法	雜湊法
演算法特性	● 資料不須排序過 ● 適用動態資料 ● 演算法簡單	● 資料須排序過 ● 適用固定資料 ● 資料結構必須可任意存取	● 資料不須排序過 ● 適用動態資料 ● 具保密性 ● 可做資料壓縮
最糟情況	O(n)	$O(\log_2 n)$	O(1)
平均情況	O(n)	$O(\log_2 n)$	O(1）
比較	資料項不多時有不錯的表現	資料不均勻分佈時仍有不錯效果	是最好的演算法

範題

This problem is about hashing.
⒜What is a hashing function?
⒝What is a collision?
⒞In static hashing, what is the loading density? （交大資管）

【解】見本題型【重點說明】部分。

範題

有一個hash table，其index爲0到5，如果用以下hash function以及linear probing的方式將以下的數字：32, 17, 9, 26, 83, 24放入此hash table，則此table的內容從index 0到5依序爲何？（假設hash(N) = N mod 6）（靜宜資管）

【解】

Index	0	1	2	3	4	5
Value	83	24	32	9	26	17

範題

What makes a good hash function? （南華資管）

【解】

一般選擇具⑴計算簡單；⑵碰撞發生頻率低；⑶叢集（cluster）現象少等優點的函數來作雜湊函數（hash function）。

範題

Which one of the following is not a method to resolving hashing collisions?
⒜Primary clustering ⒝Separate chaining
⒞Hash bucket ⒟Open addressing. （雲科大資管）

【解】(C)；

Hash bucket is not a method to resolving hashing collisions.

範題　《基本題》

用雜湊法（hash method）將下列七個數字存放在0, 1, …, 6的七個位置。

101, 186, 16, 315, 202, 572, 463

【解】

若以除數7進行除法（division method），餘數必介於0～6之間，即得 $f_d(x) = x\%7$ 。

$f_d(101) = 101\%7 = 3$

$f_d(186) = 186\%7 = 4$

$f_d(16) = 16\%7 = 2$

$f_d(315) = 315\%7 = 0$

$f_d(202) = 202\%7 = 6$

$f_d(572) = 572\%7 = 5$

$f_d(463) = 463\%7 = 1$

範題

下列有關hashing function的敘述，哪一項是錯誤的？

(A)bucket size和loading factor的大小不會影響hashing function的執行效率

(B)位數分析法（digit analysis）將鍵值中分布的最不均勻的幾個digit找出來作爲key，希望可以降低collision的次數

(C)以MOD(key, M)函數作爲hashing function，除數M必須爲質數

(D)linear probing、double hashing和hashing with chaining都是處理collision的技巧。　（中央資管）

【解】(B)；

剛好相反，以鍵值中分布的最均勻的幾個digit找出來作爲key，來降低collision的次數。