CHAPTER 8

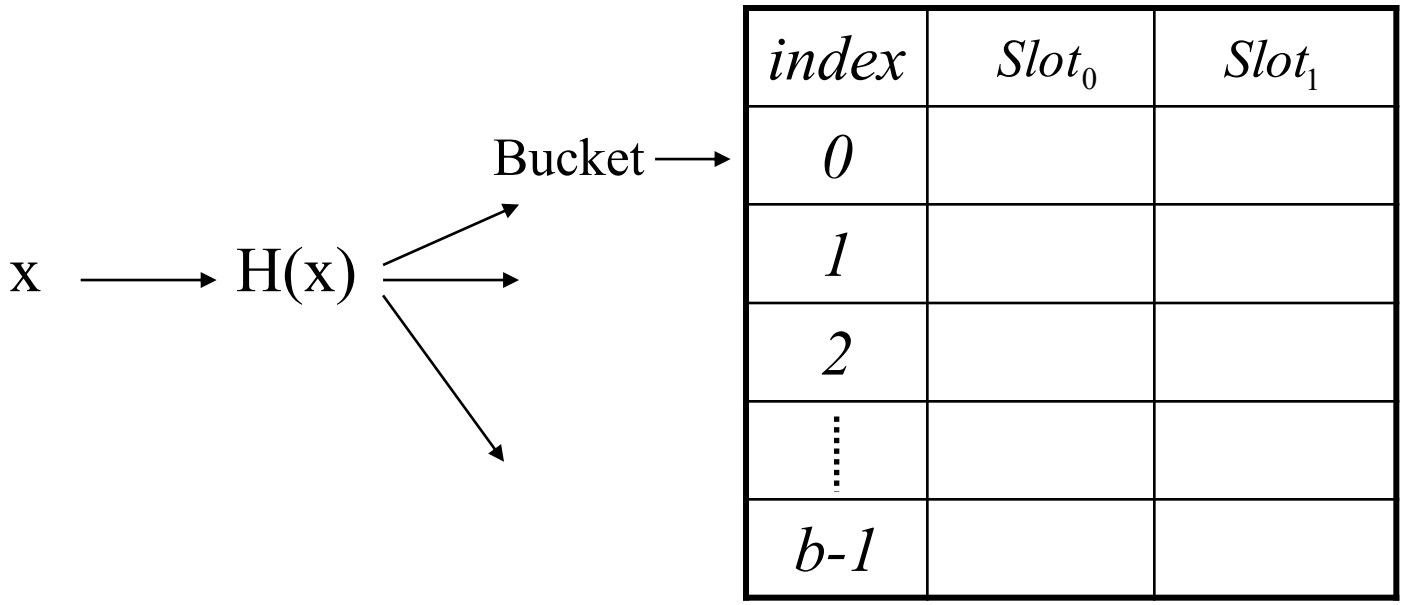
Hashing

Hashing

Definition:

為一種資料儲存與搜尋(擷取)的技術，欲儲存資料時，需先透過 Hashing function 計算，得出 hashing address (home address)，再到對應的 Bucket 中儲存或擷取資料。

Hash Table



Terminologies

Identifier density:

$\alpha \nearrow \Rightarrow utilization \nearrow \Rightarrow Collision \nearrow$

The *identifier density* of a hash table is the ratio *n/T*, where n is the number of identifiers in the table. The *loading density* or *loading factor* of a hash table is $\alpha = n/s \cdot b$ (***T***: distinct possible value of identifiers. ***s***: number of slots per bucket. ***b***: bucket number)

Collision:

不同的資料經由 hashing function 計算，得出相同的 hashing address (表示這兩筆資料將儲存在相同的 bucket 中)

Overflow:

$collision \underset{\text{一定}}{\overset{\text{不一定}}{\rightleftarrows}} overflow$

當 collision 發生時，對應的 bucket 無多餘的空 slot 可供儲存資料，導致資料無法存入 hash table 中

Hashing Function

Hashing Function 之設計:

(1) **計算宜簡單**
　　盡可能降低計算上的複雜度
(2) **Perfect Hashing**
　　盡可能減少 Collision 的發生
(3) **Uniformly Hashing**
　　經由 Hashing Function 所計算出的 Hashing Address 對應到每個 Bucket 的機率應相等，不要造成局部偏重之情況

Hashing Function

常見的 **Hashing Function** :

(1) Mid-Squre

(2) Division (Modulus)

(3) Folding

(3) Digit Analysis

Mid-square

平方值取中間數:

將 Key value 平方後，取中間適當位數值為 Hashing Address

ex)

假設 Key = 8125，而 Hashing Table 有 1000 個 Buckets

$\because (8125)^2 = 660156625$

取 “156” or “015” 均可

Modulus

取餘數:

$H(x) = x \bmod m$

m的選擇: (1) 盡量不要是2的冪次方
(2) 最好是質數

Folding

折疊相加:

將 Key value 切成等長的數字片段，再將各片段相加，其和即為 Hashing Address，而相加的方式有 (1) Shift (2) Boundary 兩種

ex)

假設 Key = 12320324111220，而 Hashing Table 有 1000 個 Buckets

⇒ 123|203|241|112|20

(1) 123+203+241+112+020=699

(2) 123+302+241+211+020=897

Digit Analysis

位數值分析:

當資料為已知時，選定基底，分析所有的 Key 其不同位數值，選定較分散者

ex)

“身份證字號後三碼”即較”身高”適合

Overflow Handling

溢位發生時之處理 :

(1) Linear Open Address (Linear Probing)

(2) Quadratic Probing

(3) Rehashing

(3) Chaining

Linear Probing

當 H(x) 發生 overflow 時則循序 (H(x)+1, H(x)+2…) 往下搜尋，直到發現空的 Bucket 或無任何空的 Bucket 為止。

ex) 一 Hash Table 有 11 個 buckets (編號 0~10)，每個 bucket 只有一個 slot，今 hashing function 為 H(x) = x mod 11，且採 linear probing 的方式來處理 overflow，則下列資料依序置入此 hash table 後其內容為何？
26, 11, 4, 15, 6, 19, 22, 10

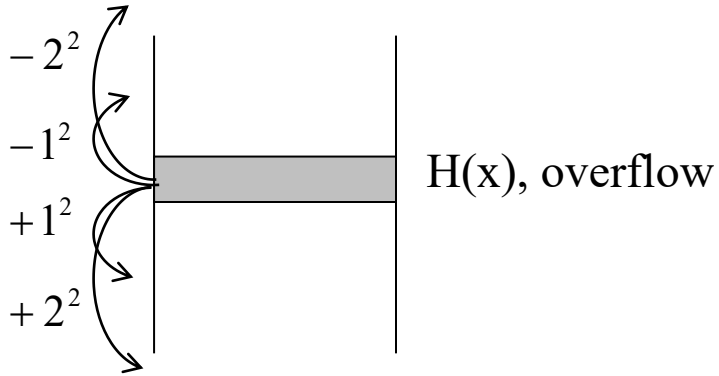
Linear Probing

優點: Simple，容易 Implement。

缺點: 易發生 clustering (群聚) 現象，造成搜尋時間增加。

Quadratic Probing

當 H(x) 發生 overflow 時則搜尋 $H(x) \pm i^2 \bmod b$ 的位置，$1 \le i \le \frac{b-1}{2}$



ex) 一 Hash Table 有 11 個 buckets (編號 0~10)，每個 bucket 只有一個 slot，今 hashing function 為 H(x) = x mod 11，且採 linear probing 的方式來處理 overflow，則下列資料依序置入此 hash table 後其內容為何？

26, 11, 4, 15, 6, 19, 22, 10, 33

Rehashing

提供一系列的 Hashing Function: $f_1, f_2, f_3 \cdots f_m$，若發生 overflow 則使用下一種函數直到沒有 overflow 發生或所有的函數皆用完為止。

Chaining

將具有相同 Hashing Address 的資料，以 link-list 的方式串接起來。

ex) 一 Hash Table 有 11 個 buckets (編號 0~10)，每個 bucket 只有一個 slot，今 hashing function 為 H(x) = x mod 11，且採 linear probing 的方式來處理 overflow，則下列資料依序置入此 hash table 後其內容為何？
26, 11, 4, 15, 6, 19, 22, 10, 33

清大通訊88) Assume that a hash function has the following characteristics:

keys 257 and 567 hash to 3

keys 987 and 313 hash to 6

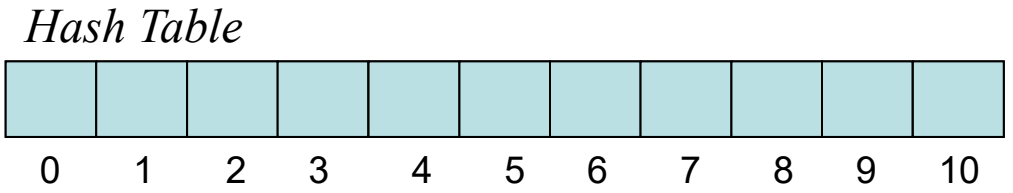
keys 734, 189 and 575 hash to 5

keys 122 and 391 hash to 8

Assume that insertions are done in order 257, 987, 122, 575, 189, 734, 567, 313, 391

(1) Indicate the position of the data if open probe addressing is used to resolve collision

(2) Indicate the position of the data if chining with separate lists is used to resolve collision



中山電機89) If H(x) = x mod 7 and separate chaining resolves collisions, What does the hash table look like after the following insertions occur: 8, 10, 24, 15, 32, 17?
Assume that each table item contains only a search key.

成大資工90) Please define hash searching, and how do you resolve the collision problem? Give examples to illustrate your answer