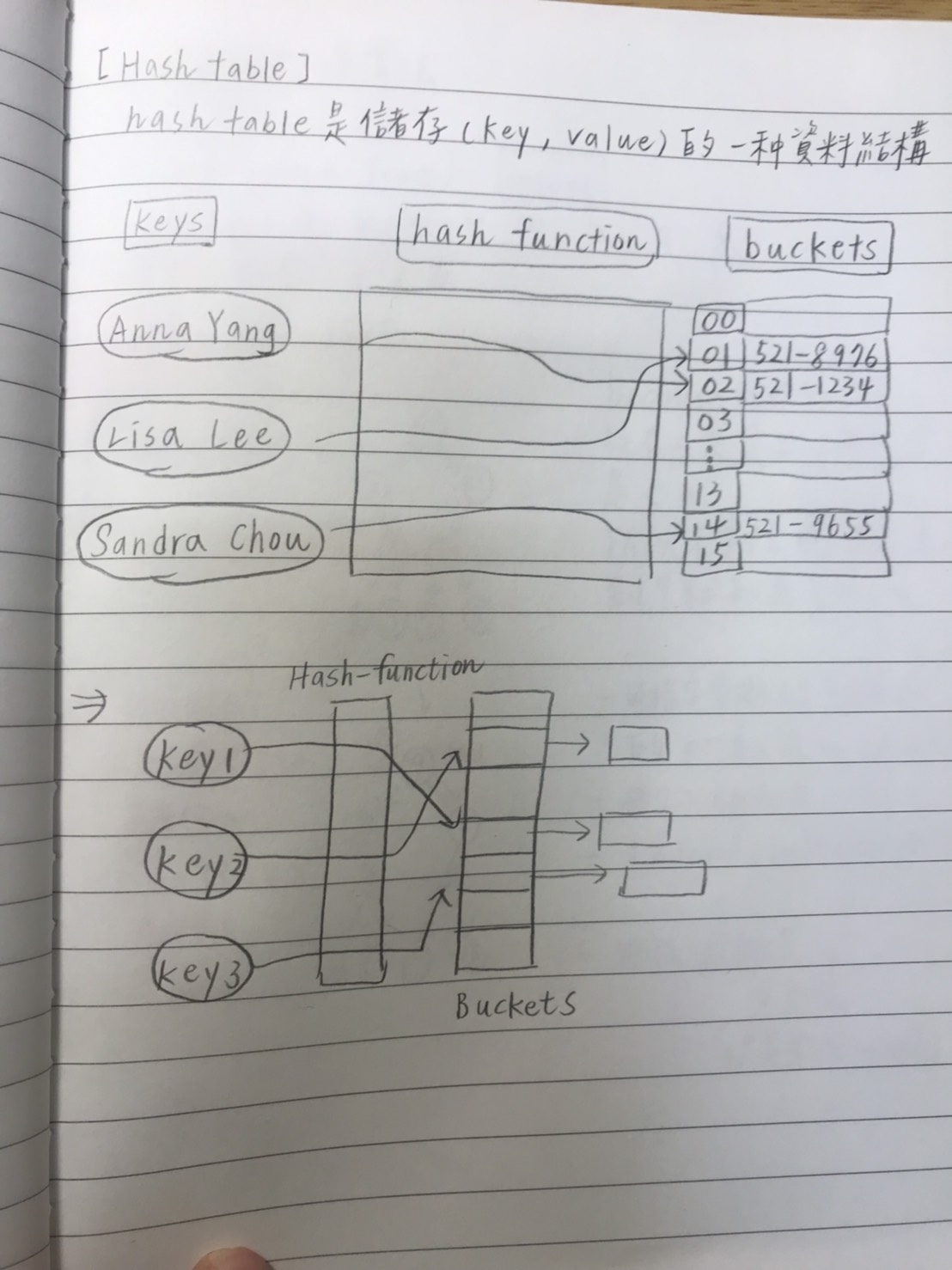
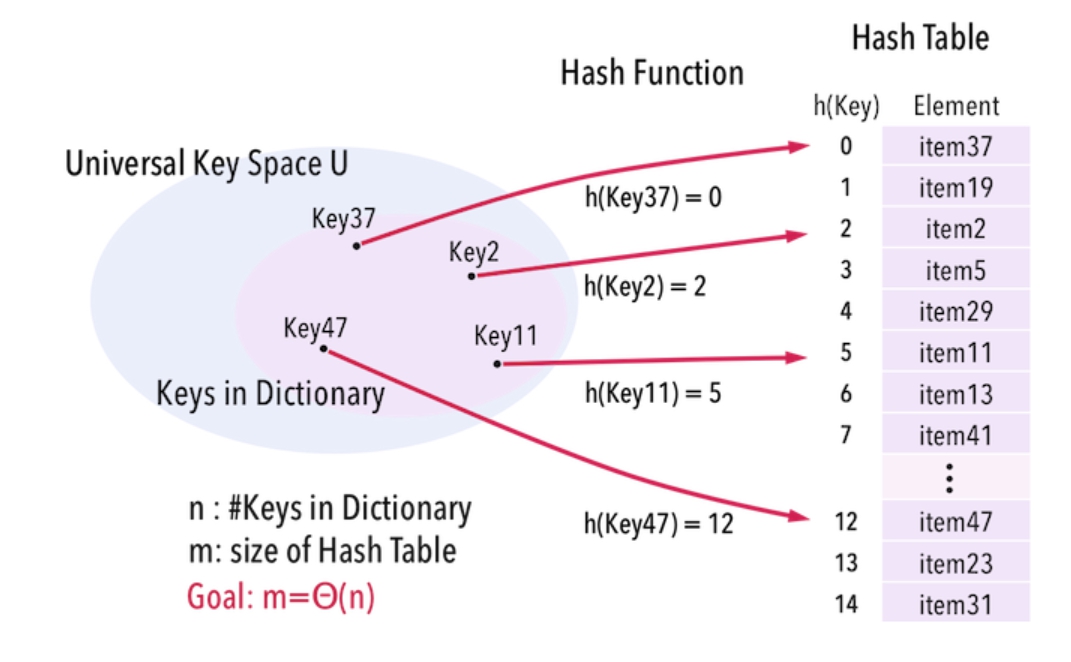
Hash Table流程圖、學習歷程與Hash Table與Hash Function原理

【流程圖】



【Hash Table原理】

Hash Table 是儲存 key,value這種mapping關係的一種資料結構。就是用Hash函數運算出來的值，根據key來儲存在數據結構中。存放這些值的數組就稱為Hash Table。Hash Table主要是希望能夠將存放資料的Table大小調到「真正會存放進去Table的資料數量」，也就是「有用到的key數量」。

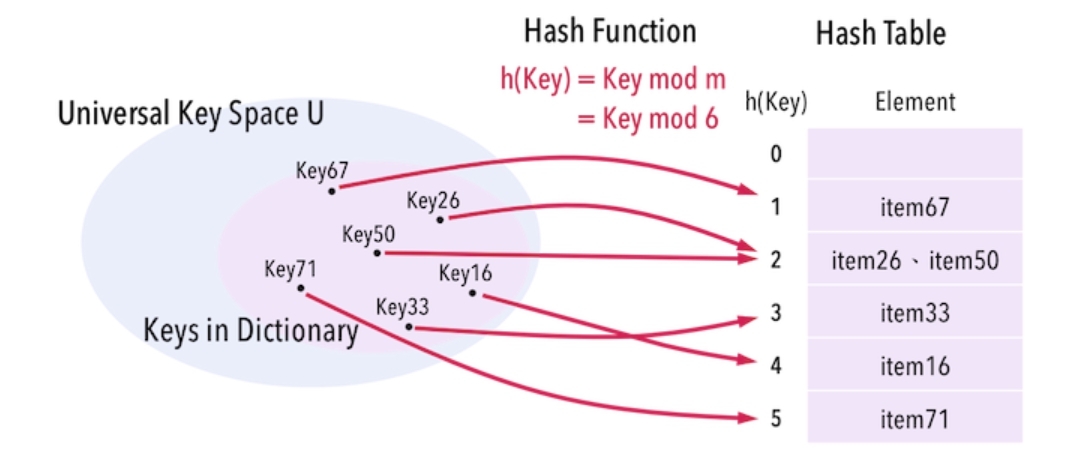


流程圖那邊也有類似的圖。

總而言之，Hash Table就是將key(像老師給的範例是用字串)，我們通過Hash Function然後把他們存到這個類似抽屜的資料結構，這就是Hash Table。

但是，

Hash Table很有可能發生Collision，簡單來說Collision就是兩筆資料會存進同一個Table之slot的情形，這樣會讓查詢資料失敗。舉例來說，我們要item1的key但是卻回傳item2的key。



用這張圖來說就是key26和key50透過hash function後都要存進Hash Table[2]

這樣就是Collision。

那麼，要如何解決這樣的問題

1. 我們可以利用Linked List把Hash到同一個slot的key串起來(老師教這個辦法)
2. 利用ProbingMethod來尋找Table中的空slot來存放資料

ps.slot就好像是table裡面的抽屜

【Hash Function原理】

Hash Function是將不定長度訊息的輸入，用演算法將他演算程固定長度的Hash值的輸出，而且所算出來的Hash值必須符合兩個主要條件：

1. 由Hash Value是無法反推出原來的訊息
2. Hash Value必須隨明文改變而改變

這樣有點深奧難懂，舉例來說，Hash Function就像是一台果汁機，我們把火龍果還有蘋果(data)，利用果汁機(Hash Function)打一打，出來的是火龍果蘋果汁。這個火龍果蘋果汁是獨一無二，而且沒有辦法變回原來的火龍果和蘋果。

之後如果我們把火龍果換成香蕉，打出來的汁也會改變，也就是說經過Hash Function後的值也會跟著改變。

Hash Function其實有很多演算法，我舉例一種老師說的演算法：

1. Division Method

(Division Method)

就是把大範圍的資料對應到小範圍的裡面，最簡單的方式就是利用取餘數的方式。假設mod是8，那麼

* H(16) = 15 mod 8 = 7 放在第7個抽屜
* H(25) = 25 mod 8 = 1 放在第1個抽屜

以此類推。而此方法最大的優點就是速度非常的快，只要做一次餘數運算就可以。

這個mod可以自己去斟酌，自己去衡量怎麼樣設會比較好。

【學習歷程】

Step1 一樣也是先回家複習課堂的東西，再上網找資料自學。首先定要先知道Hash 的基本流程，再把Hash Table 和 Hash Function弄清楚。

Step 2 再把二進位、十進位在複習一下

學習網站：

Python 內建函式進位制轉換的用法(十進位制轉二進位制、八進位制、十六進位制)

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/357214/>

step 3 開始想程式碼，但是由於前面的我還沒有打下基礎，所以還在還我的技術債。

Ref:

白話hash的簡介

<https://blog.techbridge.cc/2017/01/21/simple-hash-table-intro/>

Hash Table：Intro(簡介)

<http://alrightchiu.github.io/SecondRound/hash-tableintrojian-jie.html>

[資料結構] 雜湊 (Hash)

<https://ithelp.ithome.com.tw/articles/10208884>