

study note

Date NO.

(位址)

collision 碰撞：當二筆不同的資料雜湊相同時

Overflow 溢位：當雜湊值在 Bucket 中的 slot 已滿時
又出現一個時

常見的雜湊函式 (最常見) 哈希 (亂數) |

1. 除法 (Mod / Division)
Data % x.

2. 中間平方法 (Middle square)
Data = 235
 $235^2 = 55225 \rightarrow$ 取中 3 = 522.

3. 折疊相加法 (Folding Addition)
又可以分成兩類

3-1 Shift (移位)
ex: 987586265 $\Rightarrow 987 + 586 + 265 = 1838$

3-2 Boundary (邊界)
ex: 987586265 $\Rightarrow 987 + 685 + 265 = 1937$

可以是偶/奇數段

4 數位分析法 (Digits Analysis)
電話號碼除去頭的 09 並假設只要 2 位數
那就分析一下剩的 8 位數哪個分佈比較平均
* 方法要求已知 data 內容

Overflow 處理方法

1. 線性探測法 (Linear Probing) \Rightarrow 直接找 (由上往下)

tree + heap

max heap

parent key \geq children key

* sibling 不在此限

min heap

children key \geq parent

Insert (up heap) 由下往上比

在合規加上 node，且比較大小 (往 parent 方向) 決定位置

Extract (down heap) 由上往下比

Tree

Un directed Simple graph.

2 個 node 連在一起 稱為 edges (那條線)



join 進去 (edge)

如果沒有 edge 稱為 isolated note

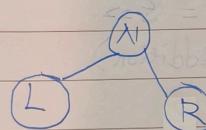
Regular graph 每個 node 連到的都一樣多線

Simple / circle path : 不同 / 相同的 node 頭尾不同

loop 連到自己的 node

Binary search trees.

- 名詞 Node (節點)：每個 tree 所連接的點。
- Root (根節點 or 樹根)：每個 tree 最初的節點 * 每個樹只有一個
- Parent (父節點)：該節點的上一層節點
- Children (子節點)：該節點的下一層節點
- Siblings (兄弟節點)：同一層且擁有共同父節點
- Leaf / Terminal (葉節點 / 終端節點)：沒有子節點的節點
- level (階層)：層數 (從最上 1 開始)
- Degree (分枝度)：該節點的總子節點數。
- Depth / height (深度 / 高度)：Tree 的總層數



DFS 先走一個子樹根再另外一個
(深度優先)

BFS 順序為 左 → 右.

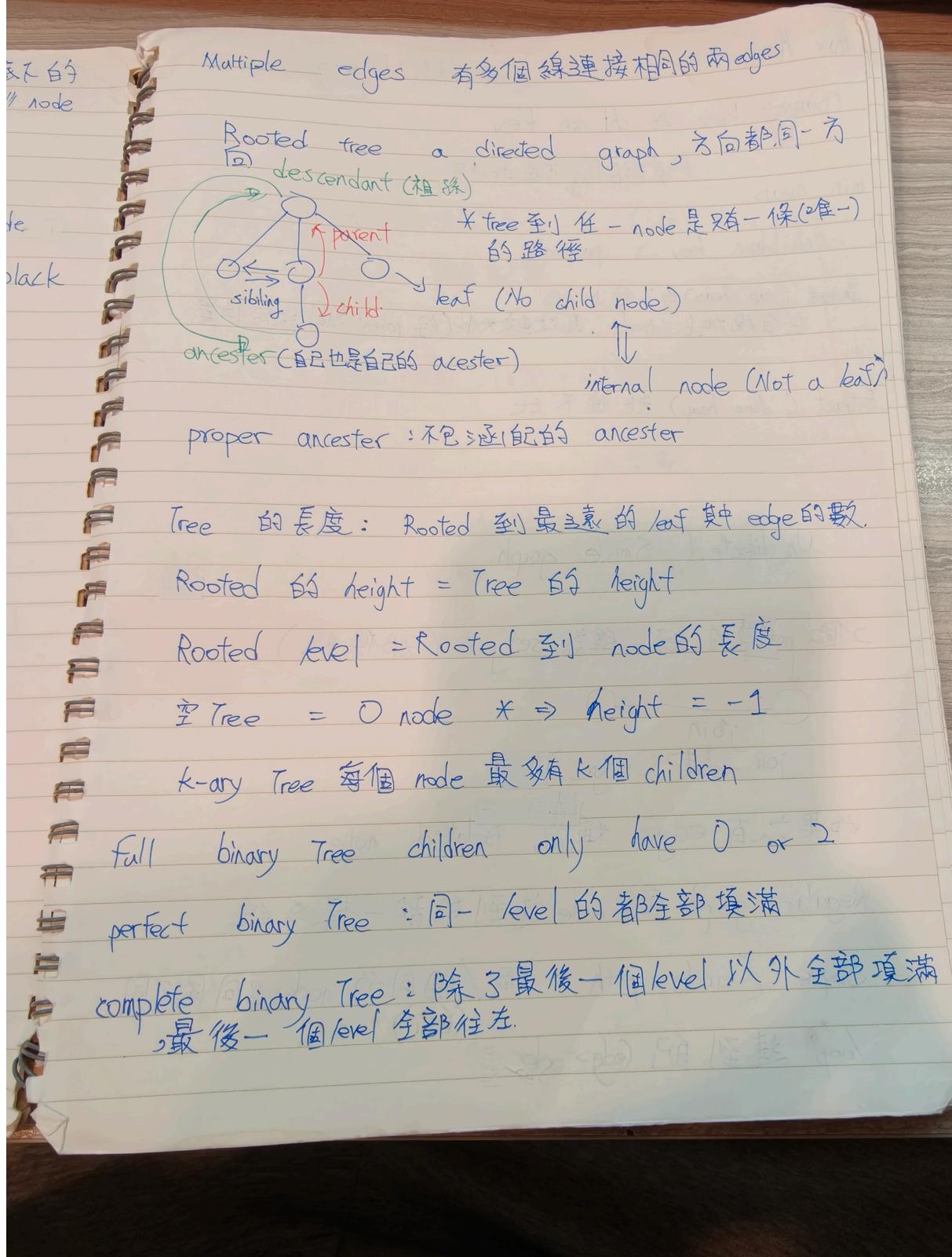
(廣度優先)

雜湊函數 (Hash function) * 將不定長度訊息的輸入，演算成固定值

將 data → [Hash function] → value

Bucket : 信儲存資料的位置

→ [✓ ✓ ✓] slot = 一個 Bucket 有幾個欄位。



red black tree

Red-black Tree * external node 最底下的 為葉子 node

rule 5:
1. Every node neither is red or black

2. root is black

3. All external node are all black.

4 在 simple path 上不可有連續二個 Red node.

5 All simple path have same number of black
number

Insertion

1. binary search 到沒有 left / right node
2. 加入 left / right node (必定為紅)

配制 Memory \Rightarrow allocated memory
 $\text{array} = (\text{int}^*) \text{ malloc } (\text{n} * \text{size of } (\text{int}))$

%P for char Resize realloc (array, size)

%d for int (void*) array Address

(void*) & array[0]

Bubble from i to h

from j to h-1

if $A[j] > A[j+1]$

ASwap