樹

Michael Tsai

2010/10/22

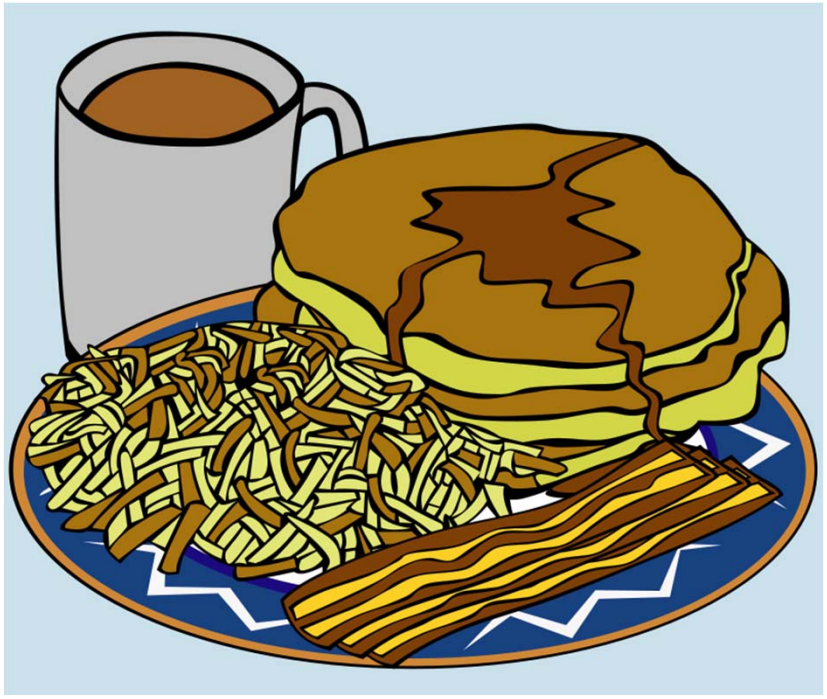
作業三今天上線

三周後要期中考拉

感謝同學們在加分題建議.
我會好好研讀+反省~

今日菜單

- 討論考試形式
- 樹
- 普通樹
- 二元樹
- 爬樹
- 堆 (啥東東)

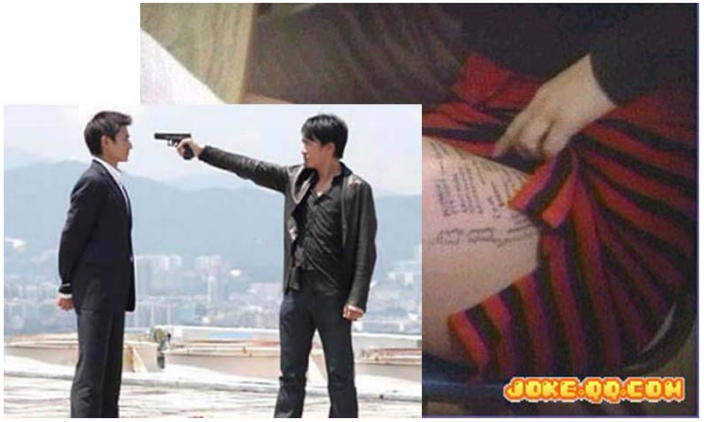


期中考!!

- 我的想法:

- 關書
- A4 大小一張, 雙面, 抄到你開心為止 (期末考沿用)
- 題目都是問答題(寫algorithm, 證明題, 問complexity)
- 請把答案寫清楚, 部分正確就有部分給分

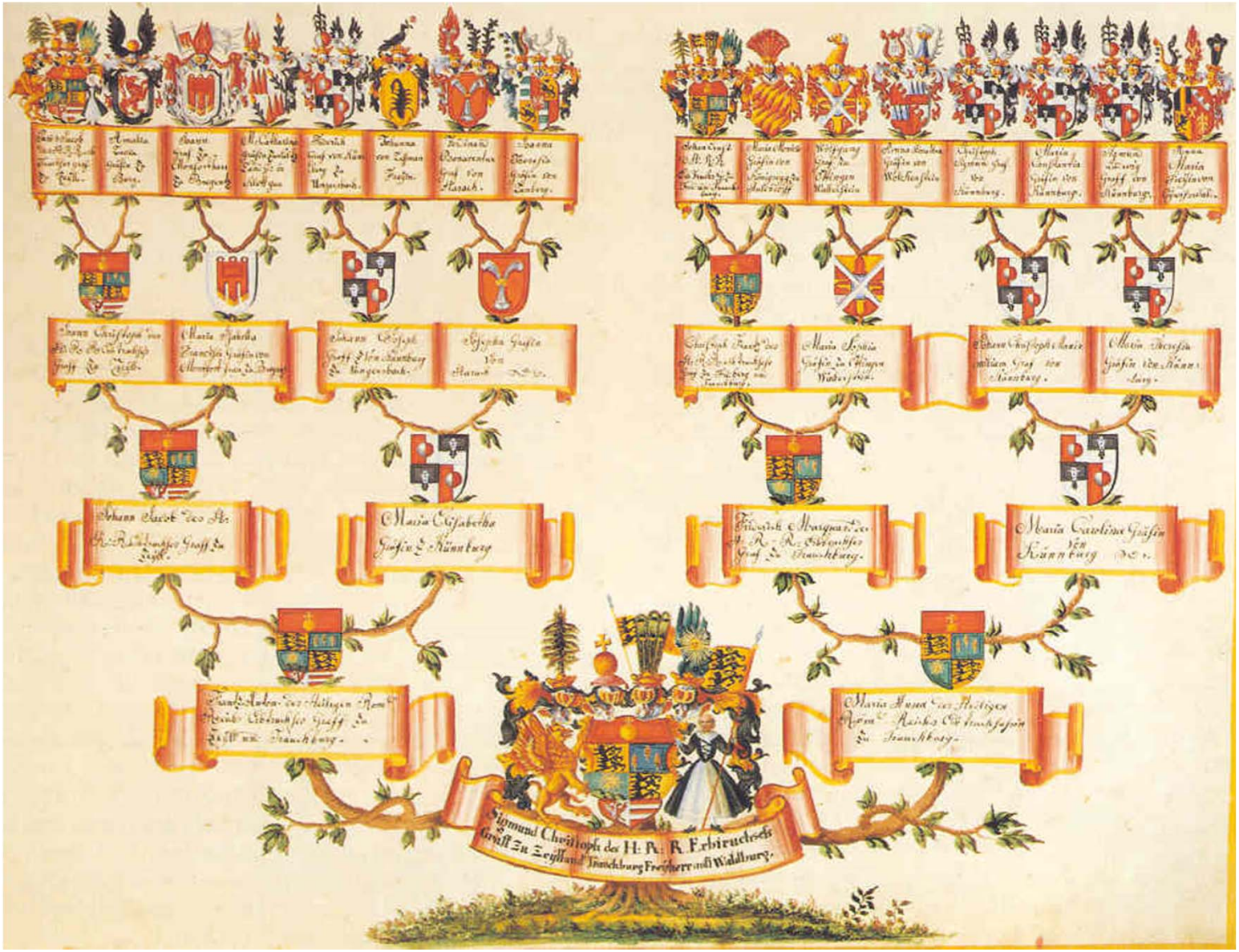
- 請發表意見.



- 作弊的直接砍頭 (當掉+送學校議處)

樹

The family tree of Sigmund Christoph von Waldburg-Zeil-Trauchburg

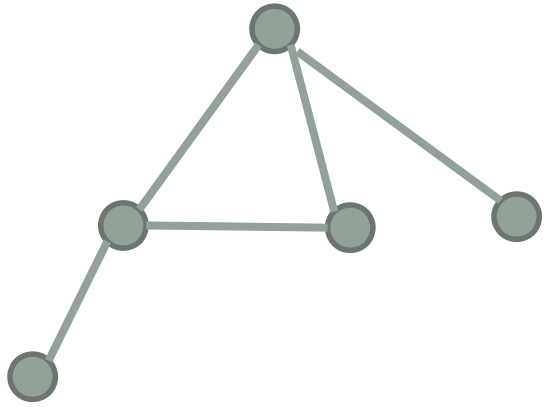


http://www.ahneninfo.com/de/ahnentafel.htm

樹的定義

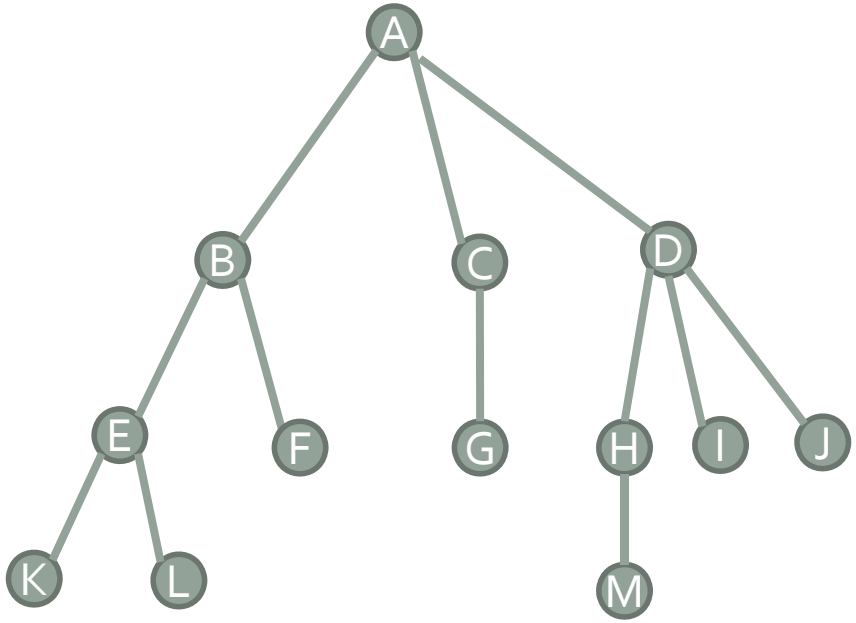
- Definition: A **tree** is a finite set of **one or more nodes** such that
- (1) There is a specially designated node called the **root**.
- (2) The remaining nodes are partitioned into $n \geq 0$ disjoint sets $T_1, T_2, \ldots, T_n$, where each of these sets is a tree.
- (3) $T_1, T_2, \ldots, T_n$ are called the **subtrees** of the root.

- 注意以上為遞迴定義
- 一個node沒有子樹的話, 是不是樹?
- 沒有node是不是樹?
- 右邊的是不是樹?
- 違反了什麼規則?



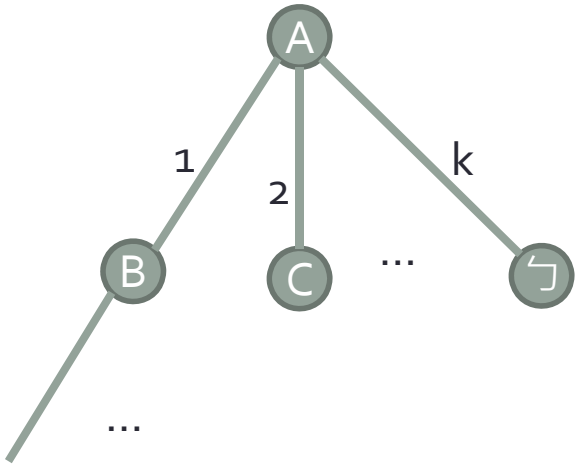
樹的字典

- Root
- Node
- Degree
- Leaf
- Terminal nodes
- Children
- Siblings
- Degree of a tree
- Ancestors
- Level
- Height or Depth



怎麼在記憶體裡面記一棵樹呢?

Data	Child 1	Child 2	Child 3	…	Child k



- 這樣有什麼壞處?
- 算算看浪費了多少空間
- 假設degree of tree = k, 總共有n個nodes
- 有多少個 child欄位是空的?
- 總共有nk個欄位
- 但是整棵樹有幾個branch? $n-1$個
- $nk-(n-1)=n(k-1)+1$

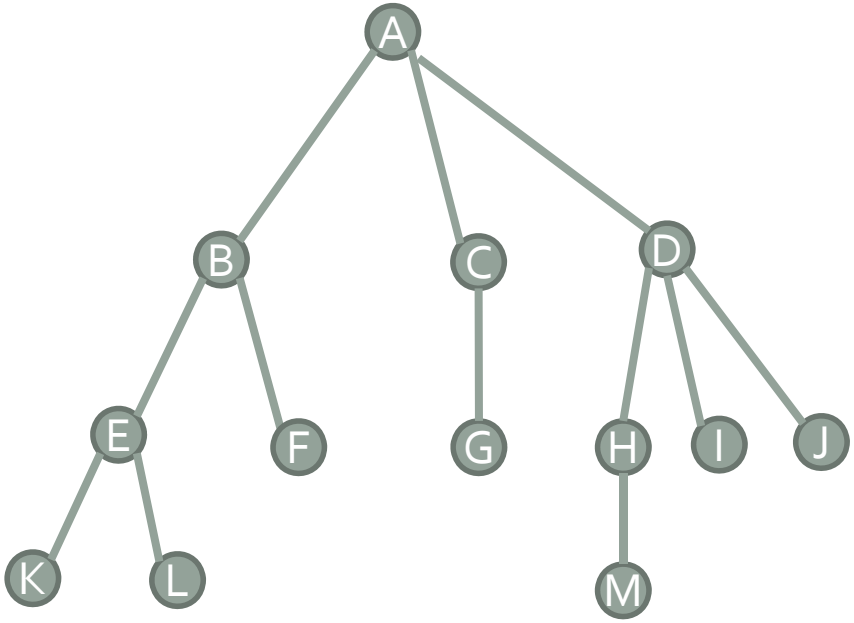
左小孩-右兄弟姊妹 表示法

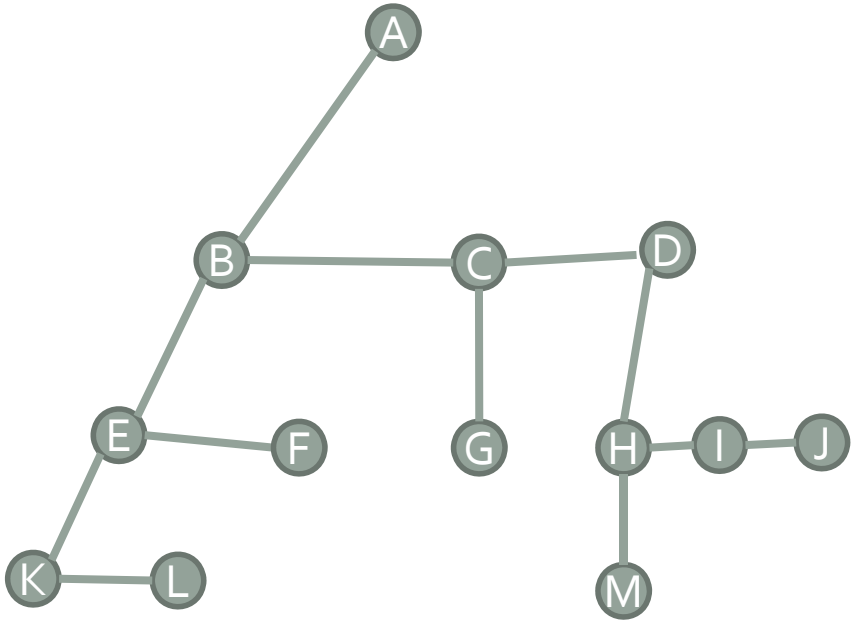
- Left child-right sibling representation

Data	left child	Right sibling

- 觀察:
- 1. 每個node最多只有一個最左邊的child (是廢話)
- 2. 每個node也最多只有一個最靠近他的右邊的sibling (也是廢話)

來畫一下 怎麼用LC-RS表示這棵樹?





左小孩-右兄弟姊妹 樹

- 可以變成 degree-two tree
- 也就是說, 是一種把普通的樹轉成degree-two樹的方法
- Root沒有右邊的child (也就是說原本的LC-RS樹裡面root不會有兄弟姊妹-廢話)

Binary Tree

- Definition: A **binary tree** is a finite set of nodes that is either **empty** or **consists of a root and two disjoint binary trees** called the left subtree and the right subtree.

- 注意: 可以是沒有node
- 比較: 一般tree不可以沒有node
- 注意: children在左邊或右邊是不一樣的 (有順序)
- 比較: 一般tree的children順序沒有差

一些證明

- 1. 在level i的node數目最多為$2^{i-1}, i \geq 1$
- 證明: 用歸納法
- i=1時, 為root那一層, 所以只有一個node, 也就是最多有$2^{1-1} = 1$個node. (成立)
- 假設i=k-1的時候成立→level k-1最多有2^{k-2}個node
- 那麼i=k的時候, 最多有幾個node?
- 因為是binary tree, 所以每個node最多有兩個children
- 因此最多有$2^{k-2+1} = 2^{k-1}$node (得證)

兩些證明(誤)

- 2. 一棵depth為k的binary tree, 最多有$2^k - 1$個node, $k \geq 1$.

- 證明:
- 利用1的結果
- 則總共node數目最多為
- $\sum_1^k 2^{i-1} = \frac{2^k - 1}{2-1} = 2^k - 1$. 喔耶.

一些證明 part 3

- 3. 對於任何不是空的binary tree, 假設n_0為leaf node數目, n_2為degree 2的node數目, 則$n_0 = n_2 + 1$.

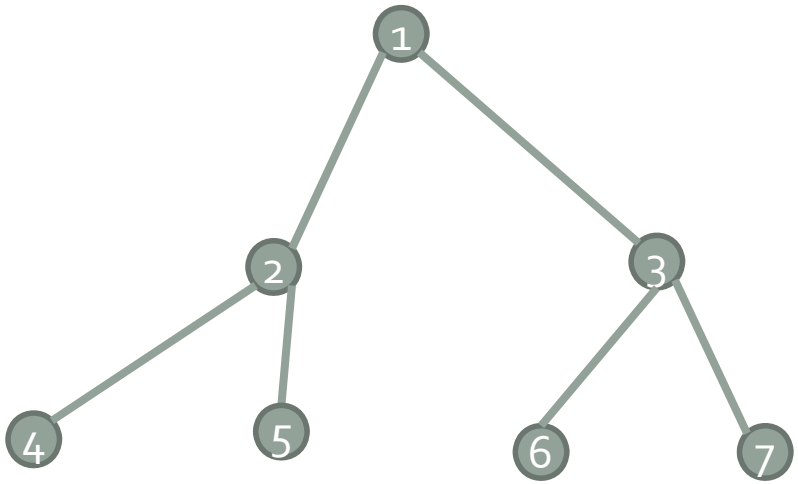
- 證明:
- 假設n為所有node樹木, n_1為degree 1的node數目,
- 則$n = n_0 + n_1 + n_2$. (1)

- 假設B為branch的數目, 則$B = n_1 + 2n_2$.(2)
- 而且$n = B + 1$ (3). (只有root沒有往上連到parent的branch, 其他的node正好每個人一個)
- $n = n_1 + 2n_2 + 1$ (4)
- (4)減(1) 得 $n_0 = n_2 + 1$. 喔耶.

Full binary tree

- Definition: a **full binary tree** of depth k is a binary tree of depth k having $2^k - 1$ nodes, $k \geq 1$.

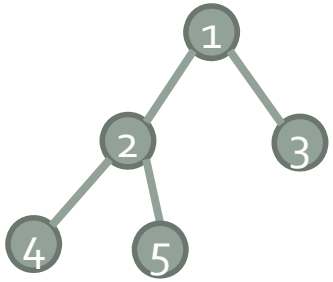
- 也就是說depth k的樹裡面最多node樹木的(滿了)



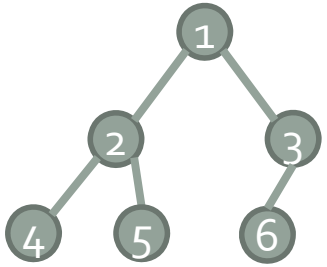
depth=3的full binary tree

Complete binary tree

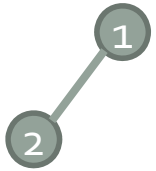
- Definition: A binary tree with n nodes and depth k is complete iff its nodes correspond to **the nodes numbered from 1 to n in the full binary tree of depth k.**



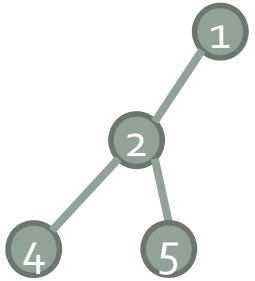
Yes



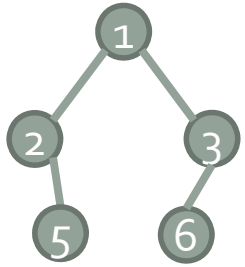
Yes



Yes



No



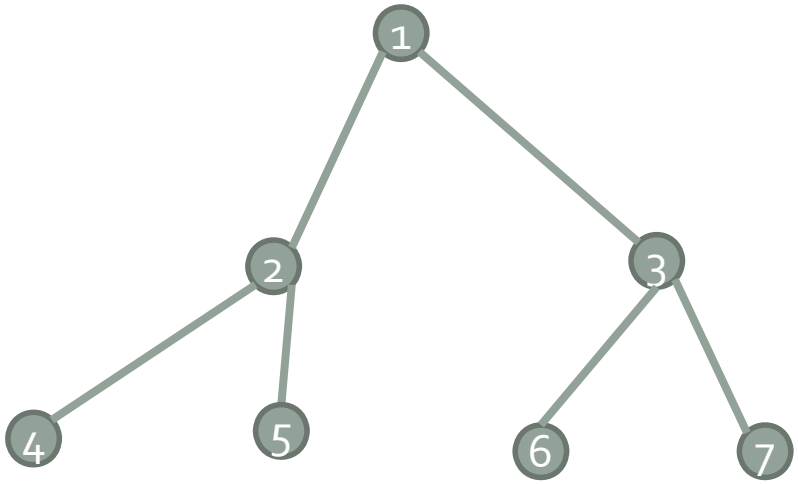
No

Complete binary tree的高度

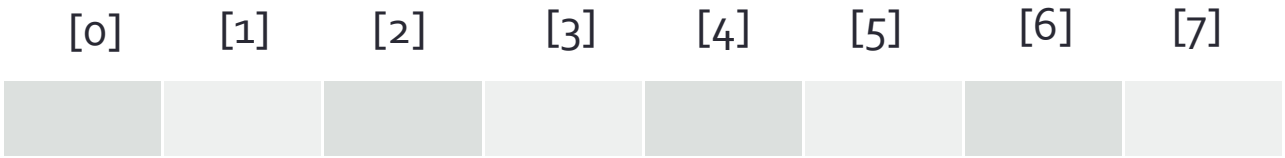
- 如果一個complete binary tree有n個node, 那麼樹的高度為?
- Hint:高度為k 的full binary tree有$2^k - 1$個node
- 答: $\lceil \log_2(n+1) \rceil$

如何在記憶體裡面表示binary tree?

- 方法一
- 提示:

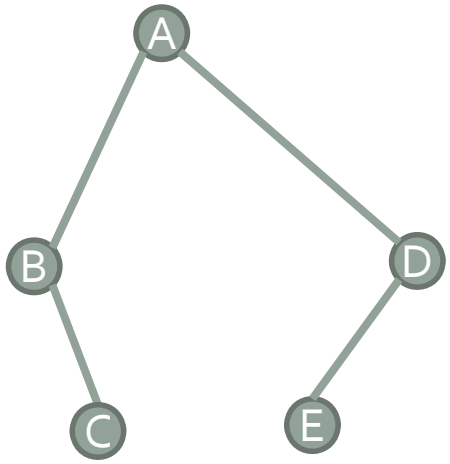


對應到:



舉例:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
空	A	B	D	空	C	E	空



壞處是什麼?

最糟的狀況浪費了多少空間?

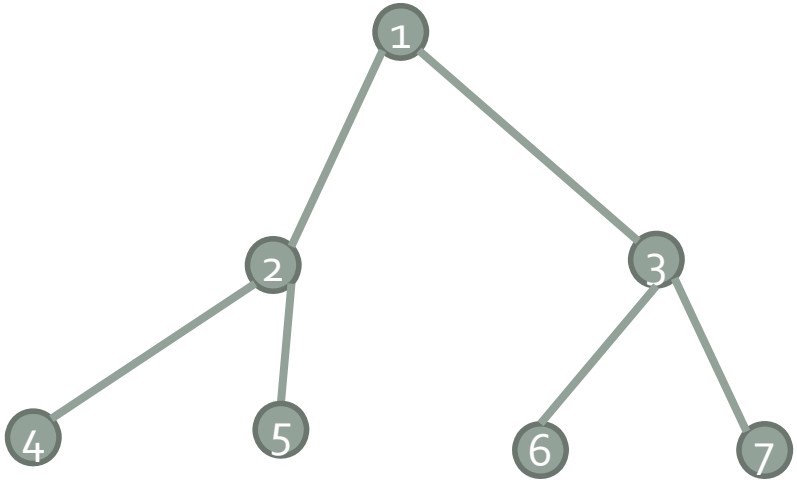
"skewed binary tree" 歪斜binary tree

規則們

- 如果有一個node 在array的index是I, parent(i)=? (index)
- 答案: $\lfloor i/2 \rfloor$

- leftChild(i)的index=?
- 答案: $2i$

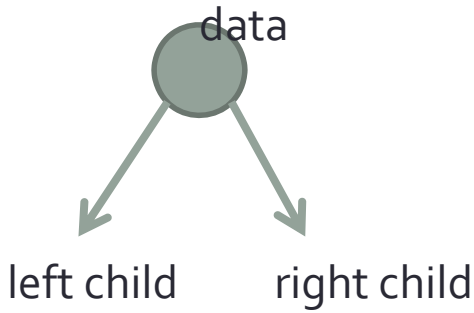
- rightChild(i)的index=?
- 答案: $2i + 1$



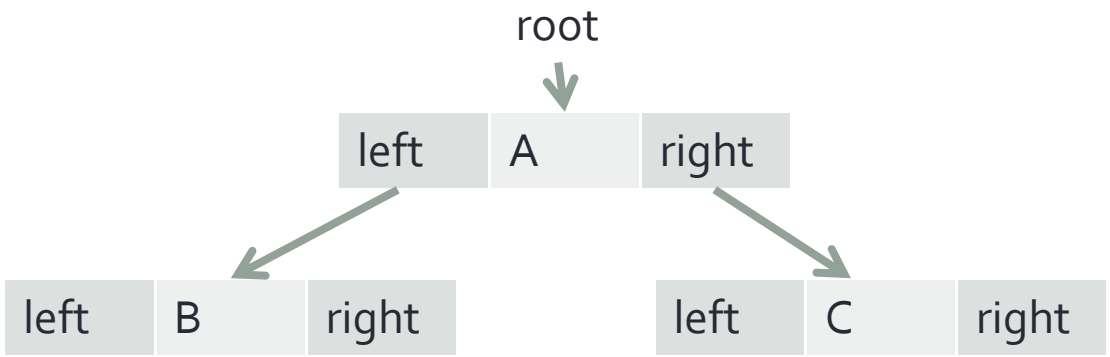
- 證明: 可以用歸納法. 請自己看課本p. 202 (Lemma 5.4)

如何在記憶體裡面表示binary tree?

- 方法2: Linked Representation

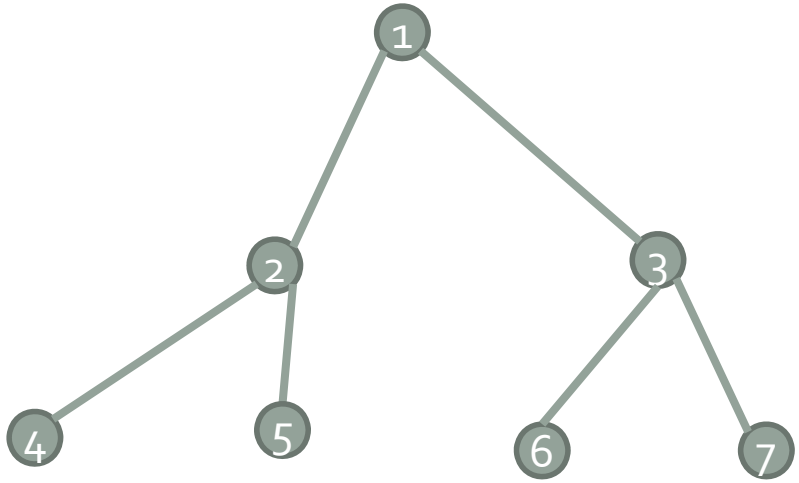


- 每個node都用malloc拿一塊新的
- 如果需要找到parent, 可以加一個新的欄位parent



Binary Tree Traversal

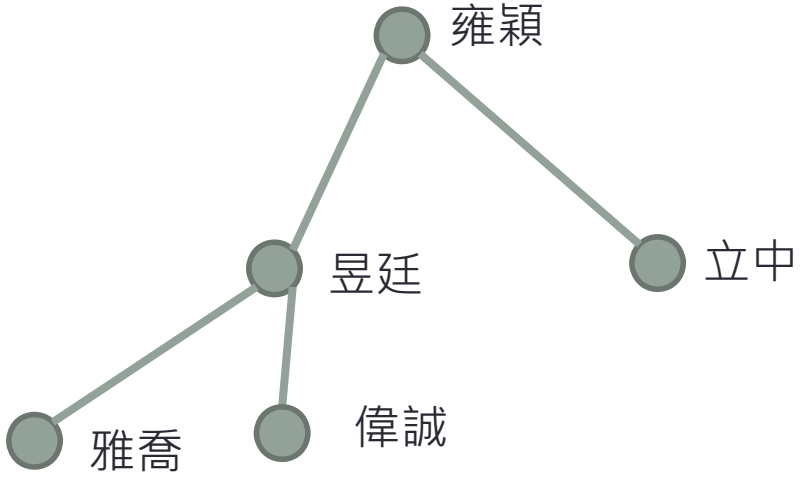
- 有一棵binary tree後, 我們要怎麼把樹的每一個node都走一遍呢?



- 到某一個node的時候, 有三件事情可以做:
- 1. 走左邊的child那邊的node們 (用L表示Left branch)
- 2. 走右邊的child那邊的node們 (用R表示Right branch)
- 3. 處理自己這個node的資料(用V表示Visit)

Binary Tree Traversal

- 如果L一定要在R之前, 那麼有三種
- VLR: preorder
- LVR: inorder
- LRV: postorder



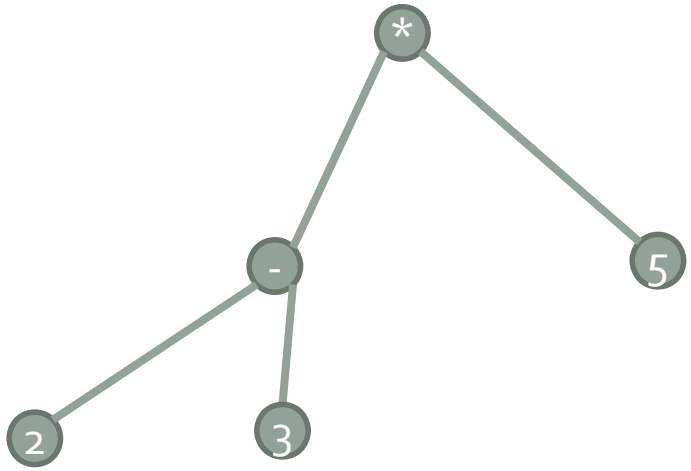
- 請同學說明preorder, inorder, postorder traversal分別順序是如何☺

Binary tree with arithmetic expression

- 每個arithmetic expression都可以建立一個expression tree

- Preorder → prefix
- Inorder → infix
- Postorder → postfix

- 請同學試試看☺
- (亂寫一個expression看看)



Binary Tree Traversal

- 可以用recursive寫法來做traversal (會很簡潔):

```
void inorder(treePointer ptr) {
        inorder(ptr->leftChild);
        visit(ptr);
        inorder(ptr->rightChild);
}
```

那如果不要用recursive寫法呢?

- 用Stack

```
for(;;) {
        for(;node;node=node->leftChild)
                push(node);
        node=pop();
        if (!node) break;
        printf("%d", node->data);
        node=node->rightChild;
}
```

Level-order traversal

- 如果改成用queue呢?

```
add(ptr);
for(;;) {
        ptr=delete();
        if (ptr) {
                printf("%d", ptr->data);
                if (ptr->leftChild)
                        add(ptr->leftChild);
                if (ptr->rightChild)
                        add(ptr->rightChild);
        } else break;
}
```

Priority Queues

- 一種每次都可以拿到priority最高的element的queue
- 直接來定義operations

- Push(element) 把element放進queue裡面
- Pop() 把element拿出來. 這個element有最高的priority

- (可以想像, 放進去的時候有做一些排序)

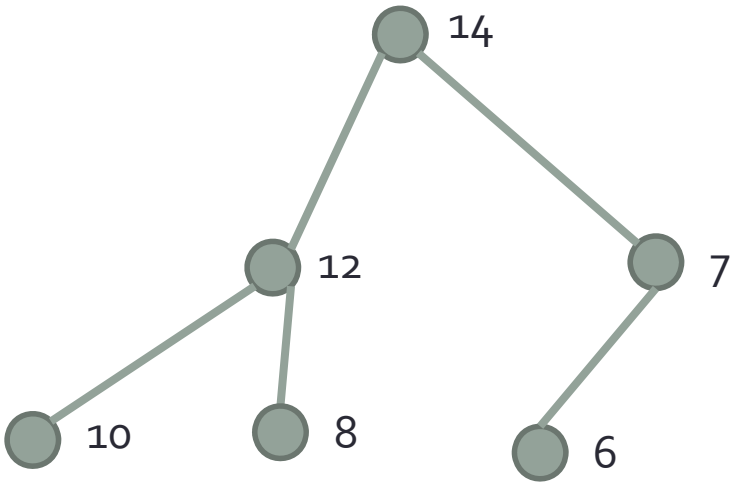
- 另外也有empty, full等等的operation
- 請同學想想看, 要怎麼用已經學過的東西來做priority queue?
- Linked List?

Heap

- Definition: A **max tree** is a tree in which the key value **in each node is no smaller (larger) than the key values in its children** (if any).

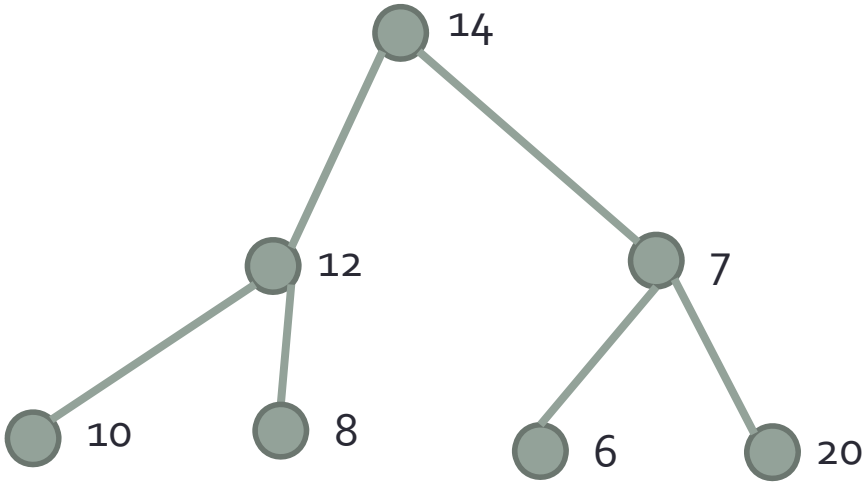
- Definition: A **max heap** is a **complete binary tree** that is also a **max tree**. A **min heap** is a **complete binary tree** that is also a **min tree**.

- 有了heap, 我們要怎麼用它來做priority queue?
- Root是不是永遠都是最大?



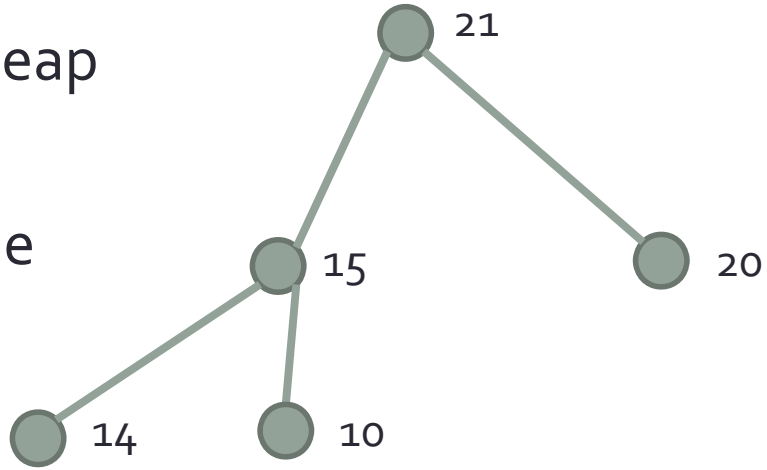
Push一個element到Heap

- 加入的時候每次都能夠繼續維持是一個max heap
- 怎麼加?
- 1. 既然是complete binary tree, 所以一定要加在下一個該出現的地方, 把新的element放在那邊.
- 2. 循序往root的方向移動, 一直到不違反parent > child的規則為止



從Heap Pop一個element

- 從root拿走一個element
- 調整位置, 繼續維持是一個max heap

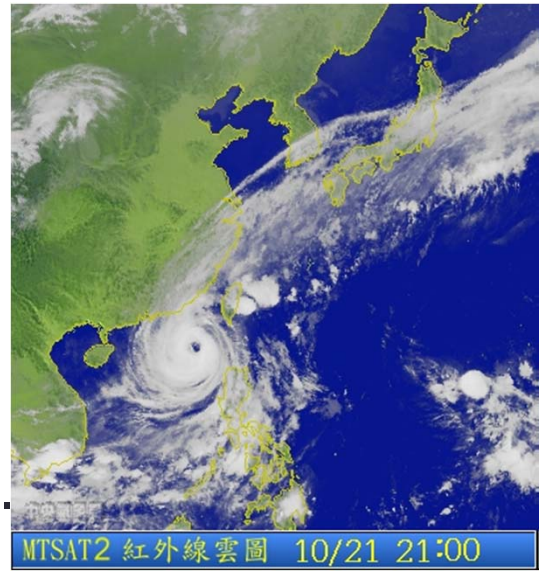


- 1. 首先既然是complete binary tree
  拿掉的位置就沒有別的選擇.
- 2. 把拿掉的位置的element,
  拿到root的地方. 和child中比較大
  的比較. 如果比其小則與其交換. 重複以上步驟直到不再違
  反parent > child的規則為止.

Heap

- Time complexity是多少??
- Push operation = O(??)
- Pop operation = O(??)

周末愉快



- 作業三今天上線
- 紙本題目有一些今天已經上過相關的部分了.
- 建議同學先看5.8.2 on p.240-243 (四頁而已嘛) :-P 才能先開始寫程式題
- Binary search tree的部分不難, 紙本題目有出, 想要先做的人可以先看5.7
- 預計下下周做考前複習. 針對同學們不懂的部分再加強.