

佇列:

1. Queue 就是排隊 → 先進先出

2. 佇列會用到以下:

1) isEmpty : 是否為空

2) enqueue : 新增

3) dequeue : 移除

4) getFront : 擷取

3. 利用堆疊和佇列可轉換前中後

4. Queue 的表示

1) 使用 A linear linked list

2) 使用 A circular linked list

My Questions

Problems & Difficulties needing exploration

Today: / /

差別: Two / One external reference

可使用佇列來運用於模擬
行為、統計、預測。

模擬:

1a. 時間

1b. 事件

My Opinions

Thoughts, inspirations, and suggestions

新到文

list
ed list

at the right time.

守時: 在對的時間, 做對的事, 來表明對別人的尊重。

《禮記》

3

密碼
cipher key

演算法

1. 分析演算法 \rightarrow 時間效率
空間效率

ii 時間效率:

\rightarrow 運算次數 \rightarrow 成長效率

以成長函數: 存在兩個常

數 k 和 n_0 使演算法 A

能在不超過 $k f(n)$ 的時

間內解決大小不少於

n_0 的問題

$$\text{Ex: } \exists k \geq n_0, (2.5n^2 - 2.5n) \leq k \cdot f(n)$$

$$\Rightarrow \exists k \geq 0, (2.5n^2 - 2.5n) \leq 3n^2$$

$$\Rightarrow O(n^2) \text{ , big } O \text{ notation}$$

My Questions

Problems & Difficulties needing exploration

* $O(n^3 + 3n) = O(n^3)$ 忽略低阶

* $O(5f(n)) = O(f(n))$ 忽略常数

* $O(f(n)) + O(g(n)) = O(f(n) + g(n))$

2. 比较与合并

→ 最多 · 平均 · 最少 :

3. 排序 : stable vs. unstable.

My Opinions

Thoughts, inspirations, and suggestions

bubble.

insertion

merge.

radix.

selection

quick.

heap.

相同值維持不變的排序

$5n \leq k \cdot f(n)$

$5n \leq 3n^2$

notation

密碼
cipher key

魔鬼躲在細節裡，天使更是這樣。

My Notes

Important Concepts worth keeping

Today: / /

My Problems &

1) bubble sort : $O(n^2)$

2) selection sort : $O(n^2)$

3) insertion sort : $O(n^2)$

4) shell sort \rightarrow ascending \leftrightarrow descending.
stable \leftrightarrow unstable

5) Merge sort \rightarrow 先分組，後合併

: 內對即外對

: $O(n \log n)$

6) quick sort : 先分組 (軸的位置)

非等分 (pivot)

後遞迴呼叫

Average case : $O(n \log n)$

Worst case : $O(n^2)$

\rightarrow 已排好

My Thought

My Questions

Problems & Difficulties needing exploration

1. radix sort :

a. 分解取部分值.

b. 過程中無排序僅搬動.

c. 需分配至對應容器.

d. 是分解後再接.

e. $O(n)$.

My Opinions

Thoughts, inspirations, and suggestions

密碼
cipher key

樹

1. 位置導向 \rightarrow binary tree

內容導向 \rightarrow binary search tree

2. 有親子關係

(a) parent of node B: A
(Above node B)

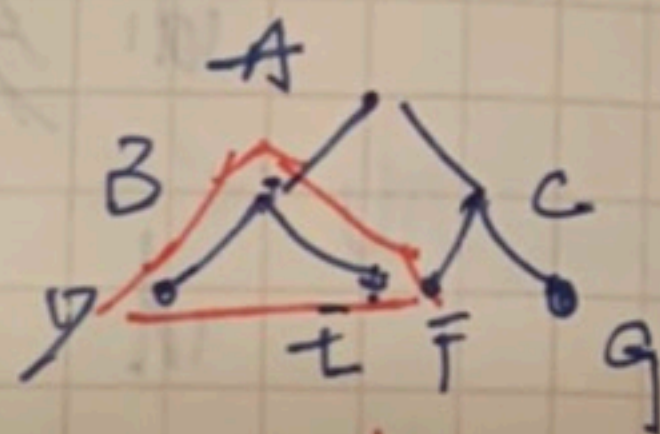
(b) child of node B: D, E
(below node B)

(c) root: no parent \Rightarrow A

(d) subtree: 子樹

(e) leaf: no child \Rightarrow D, E, F, G

(f) siblings: 兄弟: common parent



My Questions

Problems & Difficulties needing exploration

1. Ancestor of node B:

: A node on the path from root to B

2. Descendant of node B:

: A node on a path from B to leaf.

3. 二元樹，最多兩個分支。

(左子葉，右子葉)

4. Full: 當樹高 $h \Rightarrow \text{level} < h$

每個都填滿 2 個 child.

My Opinions

Thoughts, inspirations, and suggestions

Complete: 當樹高 $h \Rightarrow \text{level } h-1$ 為 full

level h , 由左至右或右至左
填滿

balanced: 每個父的左子樹和
右子樹樹高差不超過 1.

你們的話，是，就說是；不是，就說不是。

《馬太福音》

5. Complete tree ..

1. 最小樹高 $h \geq \log_2(n+1)$

$$\Rightarrow h = \log_2(n+1)$$

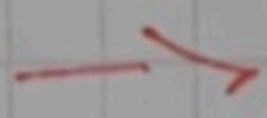
2. 最大樹高 $h \leq \log_2(n+1) + 1$

$$\Rightarrow h = \lfloor \log_2(n) \rfloor + 1$$

6. 二叉樹走訪

Traverse (In binTree : Binary Trees)

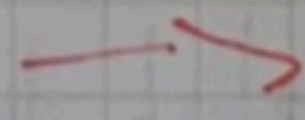
前序



If binTree is not Empty, {

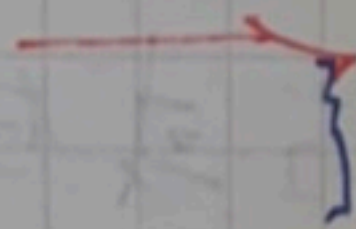
traverse (左子樹的 root)

中序



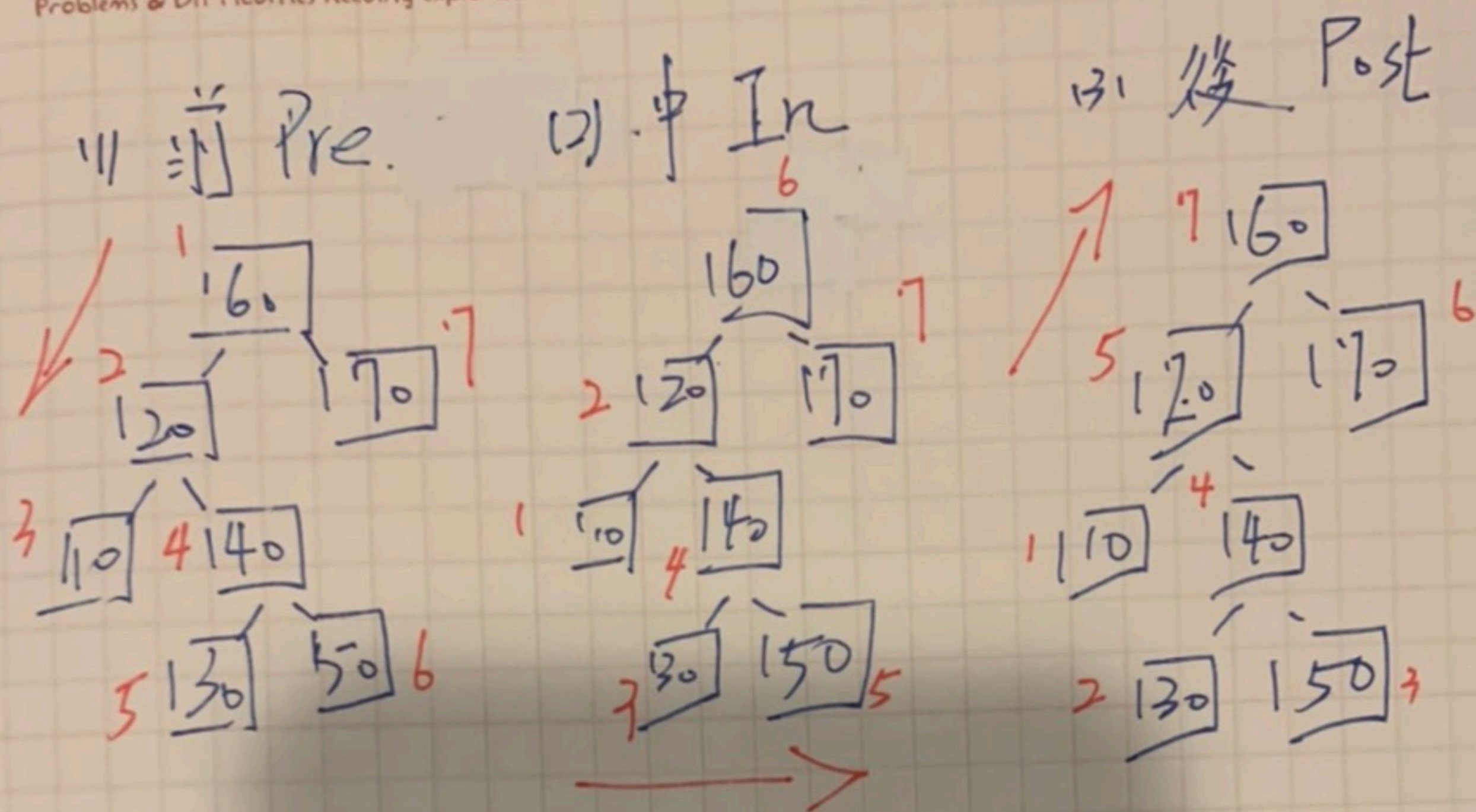
traverse (右子樹的 root)

後序



My Questions

Problems & Difficulties needing exploration



7 binary search tree.

11 值 > 左子樹的所有值

My Opinions

Thoughts, inspirations, and suggestions

12 值 < 右子樹的所有值

