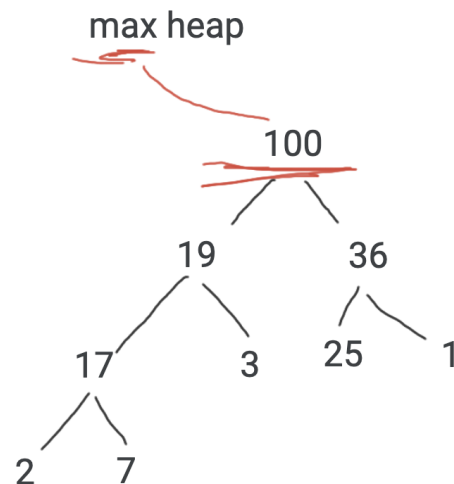


Phill-DS-0220

Heap

- tree, binary tree 不是 binary search tree → 上下有大小, 左右不分大小
- 四種 heap → max heap, min heap, min-max heap, deap
- 應用
 - heap sort → efficient $n \log n$
 - priority queue 的陽春版

Max heap

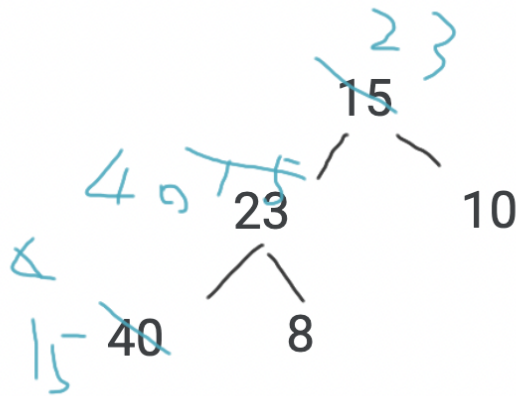


- 大小
- 排列方式 保證 最大值永遠在根節點
 - priority queue
 - $O(n \log n)$
 - sorting

建立 heap

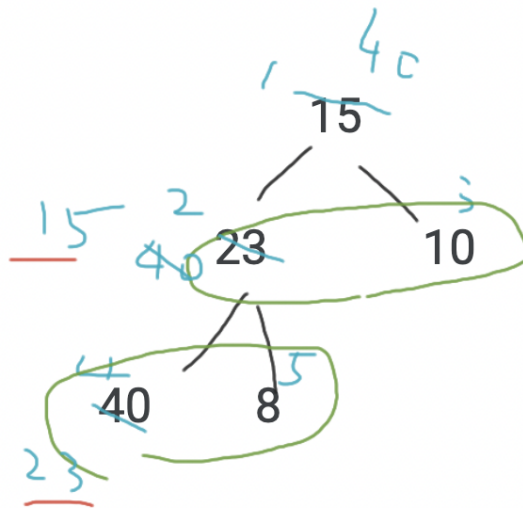
由上到下(調整)

- 鐵三角比較：上下比兩次 or 左右比再上比
- 逐層比較由上往下
- 比完再向上比一次 以免造成層與層之間大小不對的狀況



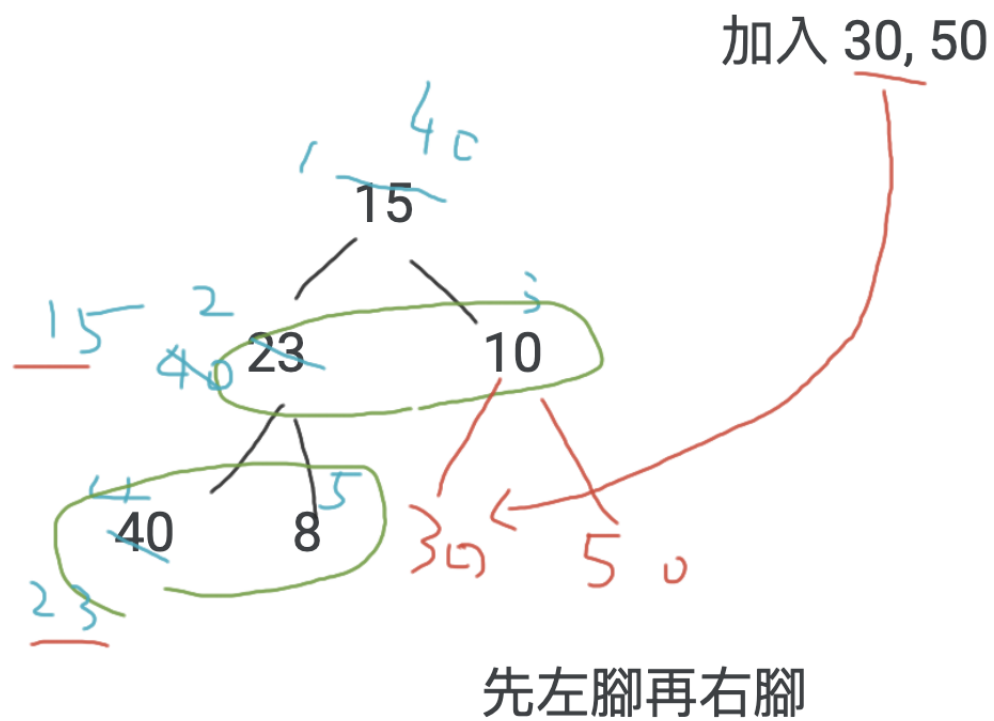
由下往上

- 對節點做編號 以大號碼開始處理
- 鐵三角同層比較 由下往上進行
- 比完以後還要向下比一次



加入元素

- 要加入的元素加入最下方
- 藉由調整的機制還原成 heap 的原則
- note: 實作上 希望先加左 再加右



刪除元素

- 刪除節點
- 先提左腳
- 轉右腳到左腳(維持先左後右的原則)

30

