Heap

一、核心概念

Heap 是一種特殊的**完全二元樹 (Complete Binary Tree)**，它必須同時滿足兩個核心性質：

1. 結構性質 (Shape Property):

- 完全二元樹：除了最後一層外，每一層都必須填滿；且最後一層的節點必須**由左至右**依序填入。

-優點：因為沒有空隙，Heap 可以非常有效率地儲存於**陣列 (Array)** 中，不需額外儲存指標。

2.排序性質 (Heap-Order Property):

根據父子節點的大小關係，分為兩類：

- 最大堆積 (Max Heap)：每個父節點的值都 ≥ 其子節點，根節點 (Root) 永遠是最大值。

- 最小堆積 (Min Heap)：每個父節點的值都 ≤ 其子節，根節點 (Root) 永遠是最小值。

二、 陣列表示法 (Array Representation)

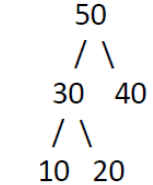
進行樹狀圖轉與陣列轉換需要運用以下公式:

- **規則**：從根節點開始，按層級由上到下、由左至右編號。

- **索引關係**（假設根節點在索引 0）：

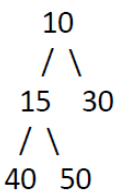
- 父節點為 i
- 左子節點為 2i + 1
- 右子節點為 2i + 2
- 任一節點 j 的父節點為 floor (j-1)/2

Max Heap



Array representation: `[50, 30, 40, 10, 20]`

Min Heap



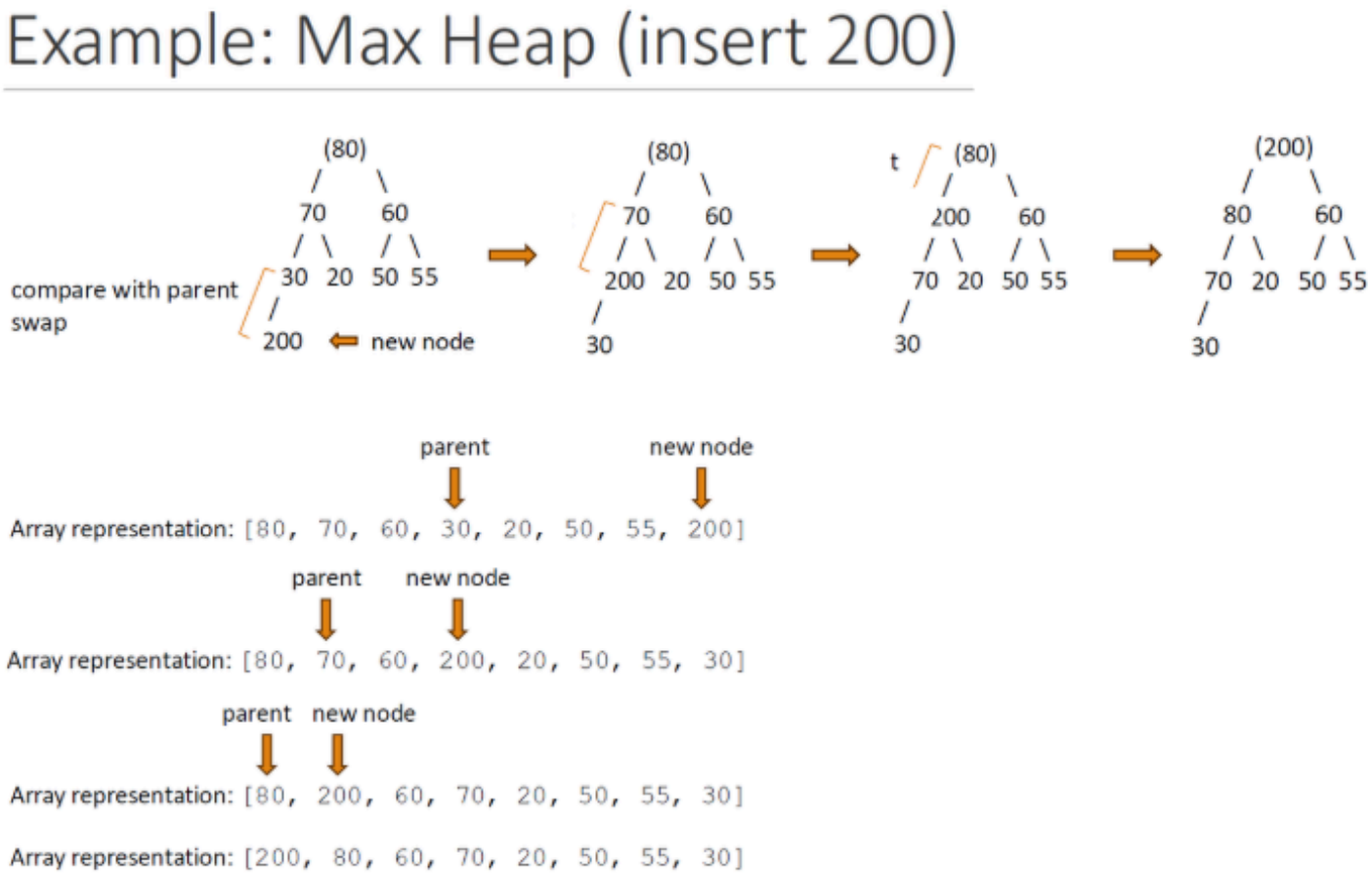
Array representation: `[10, 15, 30, 40, 50]`

三、核心操作：動態維護 Heap

Heap 最重要的功能是在插入或刪除後，樹依然是「完全」的，且順序正確。這涉及兩個關鍵動作：**Sift-up** (向上過濾) 與 **Sift-down** (向下過濾)。

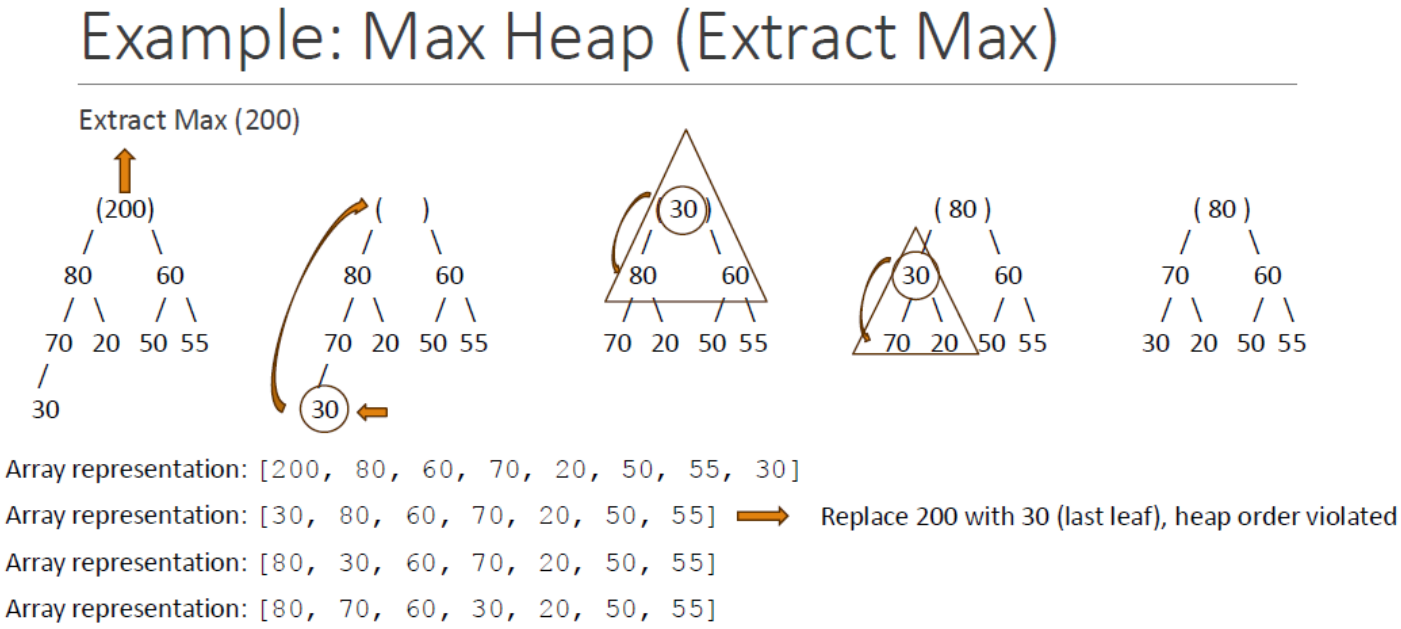
1.插入元素 (Insert) — Sift-up

1. **新增**：將新節點放在最後一層的最左邊空位（即陣列末尾），以維持完全二元樹形狀。
2. **比較與交換**：將新節點與其父節點比較。以 Max Heap 為例，若新節點較大則與父節點交換。
3. **重複**：直到滿足堆積性質或到達根節點為止。
4. **範例**：在一個 Max Heap 插入 200，它會不斷與父節點交換，直到爬升到根部。



2.取出根節點 (Extract Root) — Sift-down

1. **取代**：將**最後一個葉節點**移到根節點位置，並移除原來的根節點（最大或最小值）。
2. **比較與交換**：將新的根節點與其**較大的子節點**（Max Heap）比較，若子節點較大則交換。
3. **重複**：直到滿足性質或到達葉節點為止。



四、時間複雜度

peek_root ()	O(1)	直接看根節點
insert (x)	O(log n)	沿著樹高進行 Sift-up
extract_root ()	O(log n)	沿著樹高進行 Sift-down
build_heap (array)	O(n)	使用 Floyd's Algorithm ，比連做 n 次插入更有效率 *建立一個 Heap 的時間複雜度
heapsort (堆積排序)	O(n log n)	先呼叫 build_heap (array)，再執行 n 次 extract_root ()

五、實際應用範例

1. 急診室分級 (Emergency Room Triage)

在醫院急診室，病人並非先到先看診，而是根據病情嚴重程度分級。

- 優先權邏輯：檢傷分類等級越高（如 Level 5 代表極度危急），病人越需要被優先處理。
- Heap type：Max-Heap (最大堆積)。
- 運作方式：
  - 掛號 (Insert)：每當有新病人，護理師評估其分級後加入堆積中，時間複雜度為 O(log n)。
  - 看診 (Extract Max)：醫生永遠從 Heap 的根節點 (Root) 帶走病人，因為根節點保證是當前優先權最高（等級最大）的人。

2. 作業系統 (OS) 排程：最短作業優先 (SJF)

為了提升系統反應速度，OS 會優先執行「剩餘執行時間最短」的程式。

- 優先權邏輯：剩餘時間越小，優先權越高。
- Heap type：Min-Heap
- 運作方式：
  - 新任務：當程式發出執行請求，OS 將其「預估執行時間」放入 Min-Heap。
  - CPU 分配：CPU 核心每次從 Heap 的根節點取出數值最小 (時間最短) 的程式來執行。這樣可以確保大量的小型任務快速完成，不會被一個巨大的任務卡住。

3. 網路數據：高品質服務 (QoS) 封包處理

網路路由器在處理數據傳輸時，會面臨頻寬有限的問題。例如，語音通話 (VoIP) 的封包必須即時傳輸，而下載檔案的封包可以稍等。

- 優先權邏輯：封包標頭 (Header) 中帶有 QoS 等級，高等級的標籤 (如 QoS Level 7) 代表高優先權。
- Heap type：Max-Heap
- 運作方式：
  - 緩衝區管理：所有進入路由器的封包會先存放在一個以 Heap 結構維護的緩衝區 (Buffer)。
  - 傳送順序：當輸出鏈路有空閒時，路由器執行 peek_root() 找到優先權最高的封包發送。即使後來的封包 QoS 等級更高，它也能透過 sift-up 操作快速移動到 Heap 的頂端，確保被優先傳送。