

第3次練習-練習-PC3

學號：112111207

姓名：陳品霖

作業撰寫時間：60 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2025/01/05

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

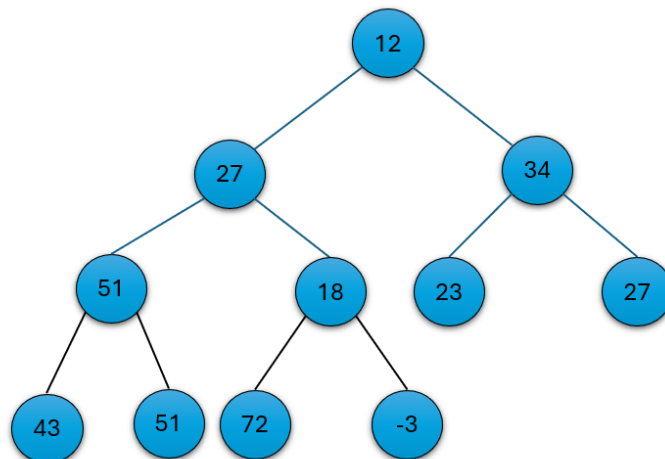
- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

1. (請參照題目pdf)

Ans:

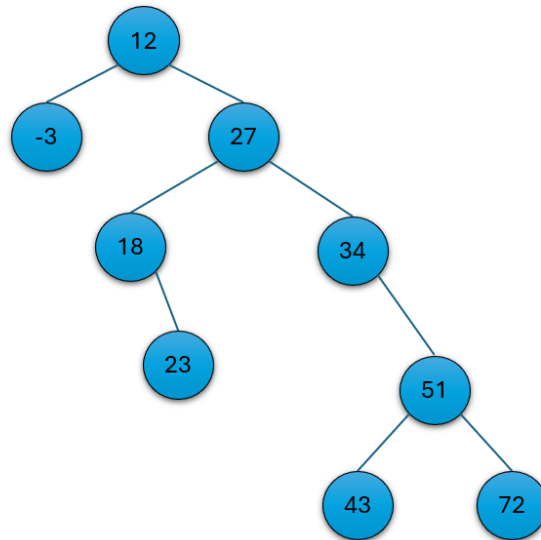
畫出二元樹(binary tree)

資料序列：12、27、34、51、18、23、27、43、51、72、-3。



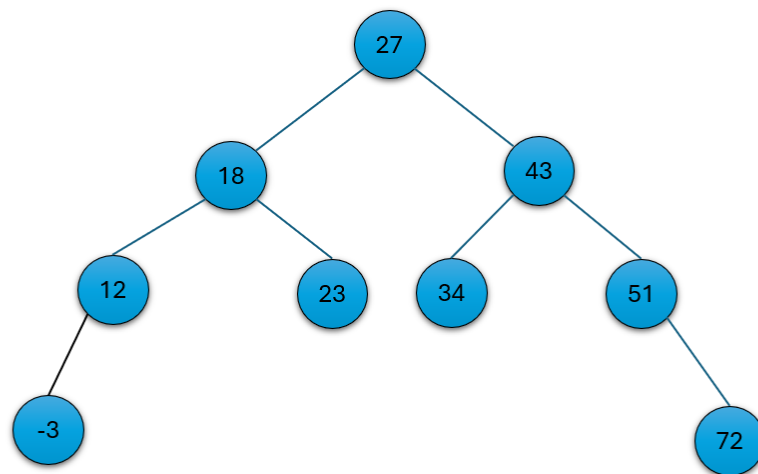
二元搜尋樹(binary searched tree)

每個節點左邊的值要比右邊小(重複的值略過)

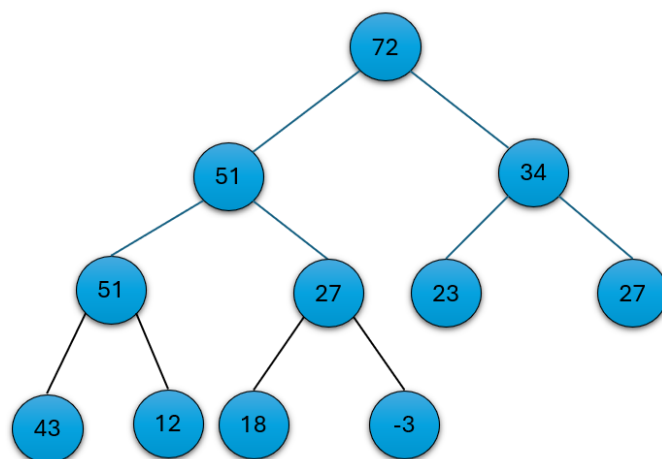


AVL Tree(AVL樹)

平衡因子確保不超過1



最大堆積樹(Max Heap Tree)



2. (請參照題目pdf)

Ans:

在堆積樹 (heap tree) 中，堆積化 (heapify) 是指將一個無序的二元樹調整為滿足堆積性質的過程。這個過程是堆積資料結構的核心操作，用來維持堆積的性質。

堆積化 (Heapify) 的定義

- 堆積性質：
 - 最大堆積 (Max-Heap)：每個節點的值大於或等於其子節點的值。
 - 最小堆積 (Min-Heap)：每個節點的值小於或等於其子節點的值。目標：調整子樹中的節點，使其成為一個合法的堆積 (即滿足最大堆或最小堆的性質)。

3. (請參照題目pdf)

Ans:

案例：搜尋引擎的關鍵字自動補全

問題描述：

當用戶在搜尋引擎 (例如 Google) 輸入部分關鍵字時，系統需要快速提供可能的補全建議。例如，用戶輸入「tai」，可能出現：

- Taiwan
- Taipei
- Tai chi
- Tailwind CSS



解決方案：

在這個情境中，樹狀架構（特別是字典樹，Trie）是一種非常合適的資料結構，用來管理和查詢關鍵字的自動補全。

樹狀架構特別適合用於具有層級結構、需要高效查詢或動態操作的場景。在關鍵字自動補全的案例中，Trie 是一種高效解決方案，能在節省空間的同時，提供快速的查詢能力，顯示了樹狀架構的強大優勢。

個人認為完成作業須具備觀念

完成本次作業，需具備幾個核心觀念。必須了解樹狀結構的基本概念，包括節點、根節點、子節點與層級關係。此外，還需掌握堆積化的原理，並熟悉最大堆積與最小堆積的特性。過程中，應使用二元樹的層級插入法，並結合堆積化操作，確保樹的完整性與正確性。最後，實作中須考慮時間複雜度，特別是堆積化操作的效率。