

第3次練習-練習-PC3

學號：112111225

姓名：林芷羽

作業撰寫時間：180 (mins，包含程式撰寫時間)

最後撰寫文件日期：2025/01/07

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

說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

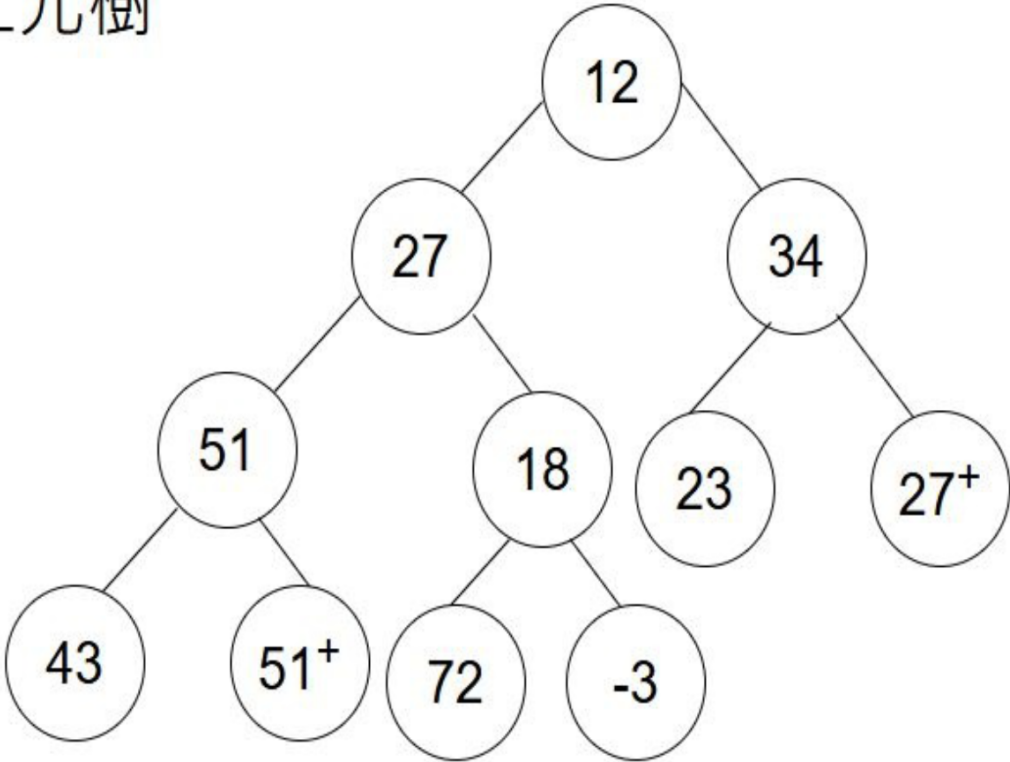
更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

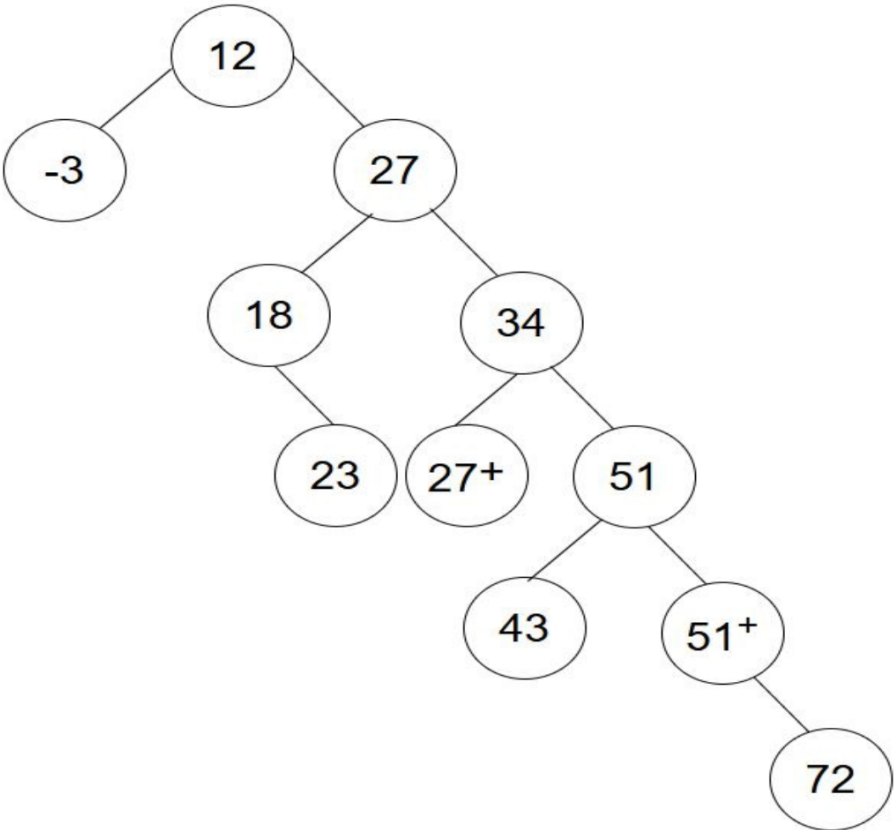
1. (請參照題目pdf)

Ans:

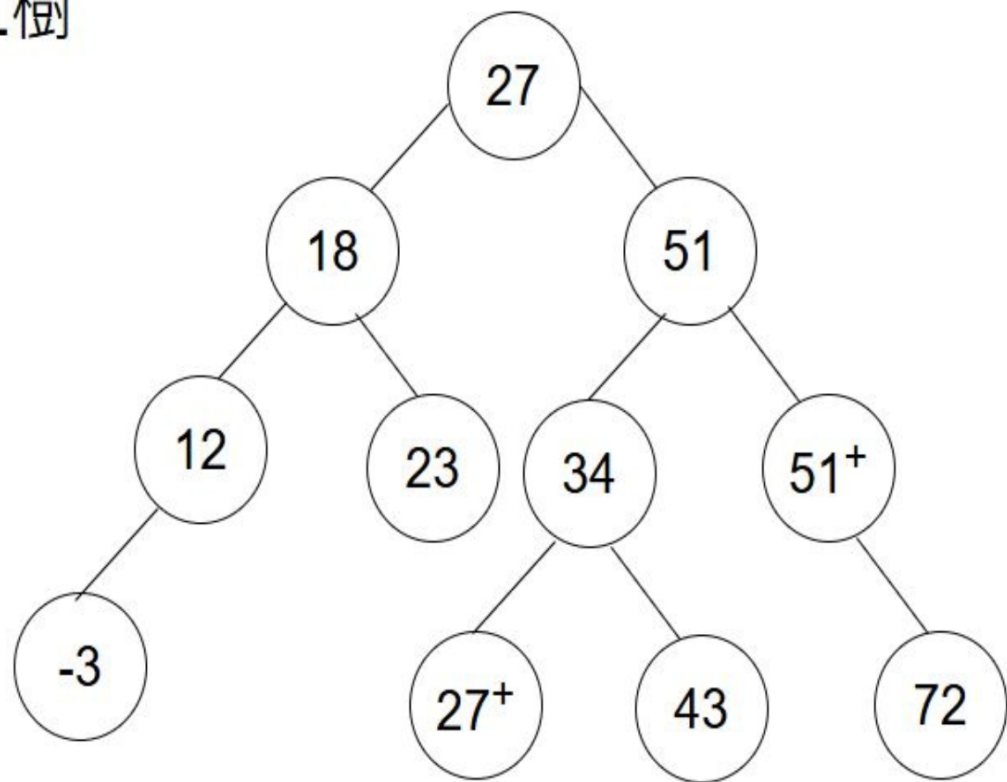
二元樹



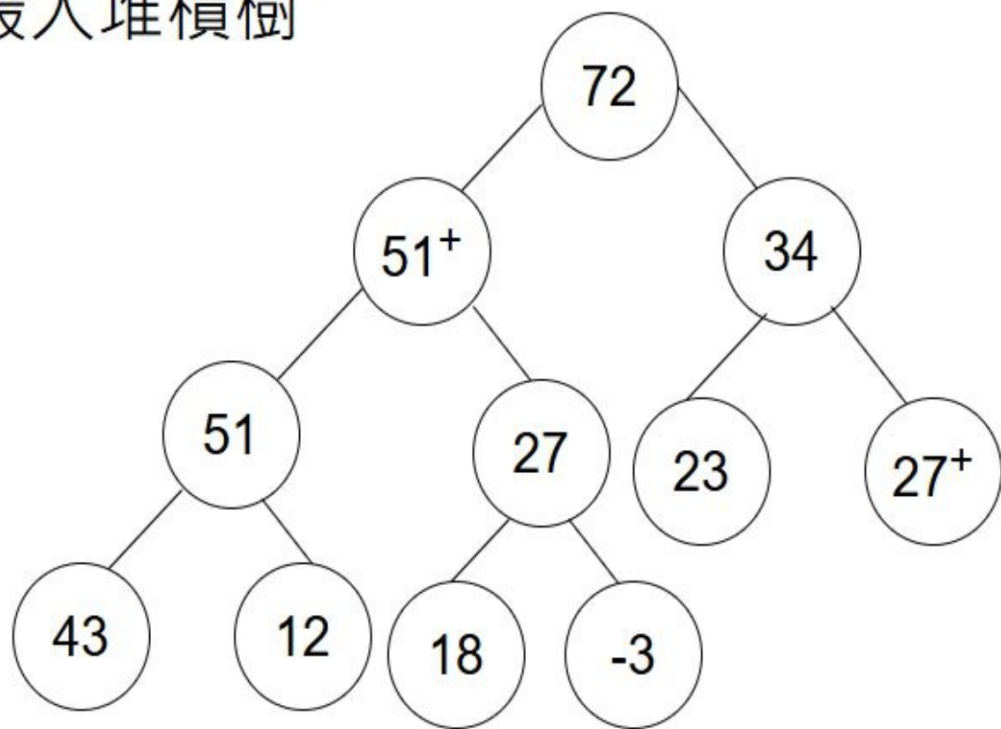
二元搜尋樹



AVL樹



最大堆積樹



2. (請參照題目pdf)

Ans:

堆積化 (heapify) 是將一個無序的樹或數列轉換為堆積結構的過程。堆積結構是一種特定類型的二叉樹，其中每個父節點的值滿足堆積規則。根據堆積類型的不同，這些規則分為兩類：最大堆積 (Max Heap) 和 最小堆積 (Min Heap)。

堆積性質

最大堆積 (Max Heap)：在這種結構中，每個父節點的值都大於或等於其子節點的值。這意味著根節點是整個樹中最大的元素，對於任何節點，父節點的值始終大於或等於子節點。

最小堆積 (Min Heap)：與最大堆積相反，每個父節點的值都小於或等於其子節點的值。因此，根節點是整棵樹中最小的元素，對於每個節點，父節點的值始終小於或等於子節點。

堆積化過程

堆積化的目的是將一個無序的數列或樹結構調整為滿足堆積性質的樹。這個過程通常是從底層的非葉子節點開始，逐步向上調整樹結構，直到整棵樹都符合堆積條件。具體步驟如下：

選擇節點：從樹的底部開始，選擇最後一個非葉子節點開始堆積化。

比較子節點：對於每個節點，堆積化操作會將其與左右子節點進行比較。根據堆積類型的不同（最大堆積或最小堆積），選擇最大或最小的子節點。

交換位置：如果當前節點不符合堆積條件（例如，對於最大堆積，若父節點的值小於某個子節點），則交換節點和最大（或最小）子節點的位置。

遞迴調整：交換後，可能會破壞堆積性質，因此需要對交換後的子樹進行堆積化，直到整棵樹滿足堆積條件。

3. (請參照題目pdf)

Ans:

範例：在線商店的商品分類系統

背景：

在線零售商，如亞馬遜、eBay 或其他大型電商平台，通常會提供數以千計的商品，這些商品會根據類型、品牌、用途等進行分類。為了便於顧客瀏覽和篩選商品，使用樹狀架構來組織這些商品是十分必要的。

樹狀架構設計：

根節點：所有商品的總分類或首頁。

這是整個分類系統的基礎，顧客從這裡開始導航整個網站的商品。

中間節點：大分類。

每個大分類代表一個主要類型的商品。例如：

電子產品

服裝

家居用品

書籍
食品與飲料

子節點：更細分的子類別。

在每個大分類下，可以進一步劃分成更具體的子類別。例如：

在 電子產品 類別下，可能有：

手機
電視
耳機
筆記型電腦

在 服裝 類別下，可能有：

男裝
女裝
童裝
運動服

在 書籍 類別下，可能有：

小說
自助書
教育類書籍
藝術書籍

葉節點：具體商品。

每個子類別下，最終會有具體的商品列出。例如：

在 手機 子類別下，會列出各種品牌和型號的手機：

iPhone 15 Pro Max
Samsung Galaxy S23
Google Pixel 8

在 男裝 子類別下，會列出具體的服裝項目：

Levi's 牛仔褲
Nike 運動T恤
H&M 襯衫

為什麼使用樹狀架構：

層級化組織：

商品有明確的層級分類，從根節點到葉節點，每一層都逐步深入，讓顧客能夠輕鬆地找到他們想要的商品。

這使得商品分類結構一目了然，顧客在瀏覽時可以快速定位到具體類別，再進一步縮小範圍至特定商品。

方便篩選和導航：

顧客在瀏覽商品時，可以根據不同層級篩選結果。例如，顧客先選擇「電子產品」，再選擇「手機」，然後選擇品牌或型號，最終篩選出自己需要的商品。

這種層層過濾的方式，能夠減少搜索的複雜度，提高效率。

適應性與擴展性：

隨著商店產品的增長，可以很容易地在樹狀結構中新增新的類別。例如，若商店開始賣家用電器，可以在「電子產品」類別下新增「家用電器」子類別。

隨著商店的發展，樹狀結構可以無縫地擴展，以包含更多類別或商品，而不會影響整體結構。

提高用戶體驗：

顧客可以根據個人需求選擇特定類別，避免了無序的商品展示，提升了整體的購物體驗。樹狀架構讓顧客不僅能夠快速找到自己想要的商品，還能輕鬆發現其他相關商品。

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結