

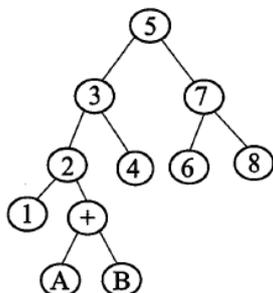
104 年特種考試地方政府公務人員考試試題

等 別：三等考試

類 科：工業行政、電力工程、電子工程、電信工程

科 目：計算計概論

一、請寫出下圖 tree 之進行後序尋訪 (postorder traversal) 的結果。



【擬答】：

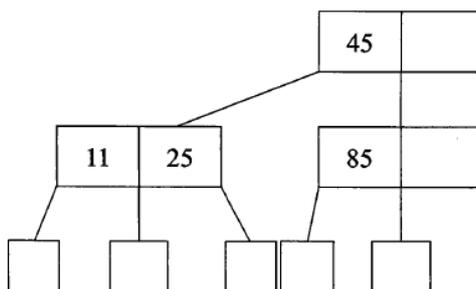
後序走訪 (Postorder Traversal)，根在後，左—右—中，得 1AB+2436875。
7—3。

二、下圖為一個 2-3tree 的資料結構：

(一)以下圖為起始之 2-3tree，畫出插入值 71 後的 2-3tree。

(二)依(一)的結果，再插入值 35 後的 2-3tree。

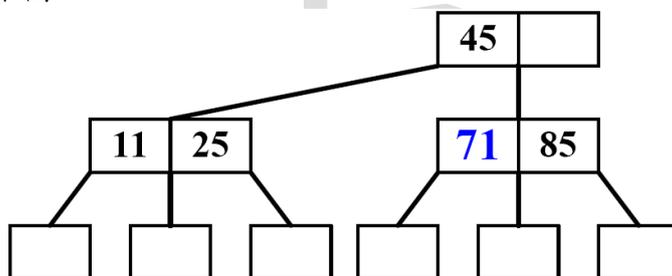
(三)依(二)的結果，再插入值 60 後的 2-3tree。



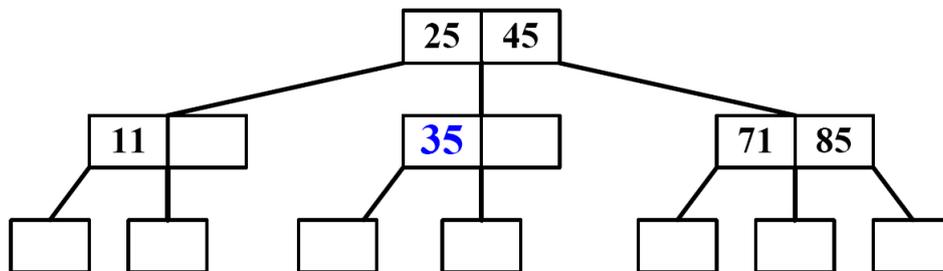
【擬答】：

三元 B 樹 (2-3 樹，最多 2 值 3 子)，高度平衡，所有葉端節點在同一層；插入索引，擺低階層 (小往左，大往右)，單值長 2 子，兩 Key 生 3 子，其餘類推，注意順序 (由小到大)；空間已滿 (索引個數 = p)，要分裂：中間值 (奇數取唯一，偶數 2 選 1) 往上拉，其餘分 2 邊 (左小右大)。

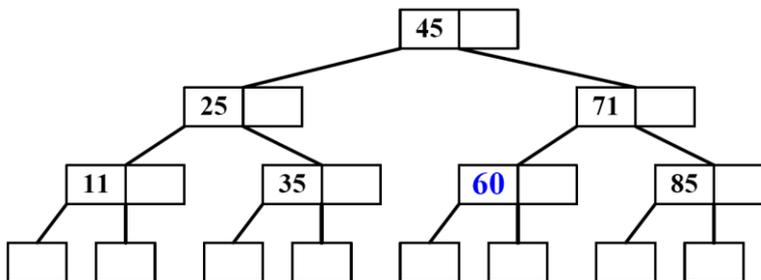
(一)插入 71：結果如下圖。



(二)插入 35：結果如下圖。



(三)插入 60：結果如下圖。



7-5。

三、對於網路連線狀況的偵測，常會使用”PING”這個指令，”PING”的使用是哪一個網路協定的應用？

【擬答】：

網際網路控制訊息協定 (Internet Control Message Protocol, ICMP)，IP 模組內之錯誤偵測與回報機制，主要功能包括偵測遠端主機是否存在、建立及維護路由資料、重導資料傳送路徑與資料流量控制；ping 指令，以 ICMP 類別 0 & 類別 8 實作之網路連線偵測工具。

8-3。

四、下列各項工作分別隸屬於 OSI 模型的哪一層？

- (一)封包路徑決定
- (二)將封包依序組合
- (三)非可靠的、非連線型的 (connectionless) 資料流傳輸服務
- (四)字碼的編碼、解碼
- (五)資料傳輸錯誤更正

【擬答】：

- (一)封包路徑決定：路由演算法，網路層。
- (二)封包依序組合：TCP 協定端對端可靠資料傳輸服務，傳輸層。
- (三)非可靠、非連線資料流傳輸服務：UDP 協定，傳輸層。
- (四)字碼編碼、解碼：內碼 (資料格式) 轉譯，表達層。
- (五)資料傳輸錯誤更正：端對端-TCP 協定，傳輸層；點對點，資料連結層。

8-1。

五、在 single instruction, multiple-cycle 的處理器設計中，執行一指令 (instruction) 需數個 machine cycle。請依序列出執行一 arithmetic/logic 指令，所需 4cycle 的步驟？在每一步驟下，詳細說明處理器 (processor) 中 control unit、arithmetic logic unit 及 registers 所完成的工作。

【擬答】：

- (一)機器週期：主要步驟有 4，擷取→解碼→執行→回存。
- (二)各步驟，CPU 內 CU、ALU 與各 Registers 工作，說明如下：
 - 1.擷取：CU 依 PC 暫存器 (Program Counter, 指向同區段內，下一個待執行指令的位址) 內容，先設定 MAR (Memory Address Register, 暫存下次讀寫的記憶體位址)，即可從主記憶體取回指令，放到 MBR (Memory Buffer Register, 暫存將寫入或已讀取的記憶體內容，又稱 Memory Data Register) 後，再轉到 IR (Instruction Register, 暫存最新擷取的指令內容)。

公職王歷屆試題 (104 地方政府特考)

2. 解碼：CU 依 IR 的指令內容，決定運算子（操作碼，OPCode）、運算元（Operand），運算結果儲存位址（Result）與下個指令位址（Next）。
 3. 執行：取回運算元，並送到 ALU 執行對應的運算動作。
 4. 回存：將執行結果回存指定位址；同時更新 PC 內容。
- 2-2。

六、物件導向程式語言的一個重要特點是「重用（reuse）」。試述「重用」的重要意義為何？物件導向程式語言提供那些語言功能使「重用」易於達成？請先列出這些功能，並加以論述。

【擬答】：

- (一) 重用 (Reusability)：重覆使用現有強健可靠類別元件，建構所需系統，可節省軟體開發時間，並降低系統複雜度與維護成本。
- (二) 物件導向三大特性：程式碼重用 3 階段，封裝（定義基本元件）→ 繼承（重用類別元件）→ 多型（修改擴充原有元件，以符合系統需求）。
1. Encapsulation（封裝）：將真實世界抽象化結果，封裝成基本類別元件；只需知道物件屬性與方法的使用方式，不需知道設計細節，能避免不當存取，保持強健性，又稱資料隱藏（Data Hiding）；主要層級有 3，能限制類別成員存取權限（Accessibility）。
 - (1) 共用：public，無取用限制。
 - (2) 保護：protected，僅類別子類別及 friend 類別（函數）內可取用。
 - (3) 私有：private，僅類別內部與 friend 類別（函數）可取用。
 2. Inheritance（繼承）：以既有類別（基底類別，Base Class | 父類別，Parent Class）+ 新的屬性或方法，建立新類別（衍生類別，Derived Class | 子類別，Child Class）；子類別 is-a 父類別（子類別具父類別所有特徵；子類別可當父類別使用），程式碼可重用，節省成本。
 3. Polymorphism（多型）：以相同名稱存取不同函式的機制，有超載（Overload）與覆蓋（Override）2 種方式，又稱同名異式。
- 4-6。