

資料結構-作業 3 報告

- 使用環境:

使用 Windows10 作業系統，並且使用 code-blocks 編譯器撰寫程式碼

- 遇到問題與困境，與解決辦法:

- ◆ PROBLEM-1

1. 遇到困境(建立完 huffman tree 之後要用對應密碼對輸入字串加密):
密碼有長有短，因為 huffman tree 是 strictly(full) binary tree 不是 perfect binary tree，所以由 root 出發給予路徑指向 leaf 的密碼大小不一，無法將每一個字元放入同樣大小的空間存取
2. 解決辦法:
原本打算使用 array 但是大小不一的情況下，放置進去若該項未存取則預設為 0，這樣會影響最後輸出密文結果，後來打算動態宣告陣列但是 compiler 顯示 conflicting types of function/initializer element is not constant，但是 malloc 語法與指標變數並無使用錯誤，所以果斷放棄改用 queue 資料結構來將各個字元 code table 放入，最後根據 traverse 輸入陣列後呼叫函數來將 queue 中，由於 rear 不斷根據輸入遞增，所以將 queue 中小於 rear 的 index 項輸出

- 程式內容:

- 1 先將建立 huffman tree 的結構跟 linked-list 指標還有 max heap tree 一一用函式寫出，建立節點與判斷大小存放機制
- 2 將需要判斷的數值加入生成 huffman code 的函數，並用先前寫的函數呼叫 root 到 leaf 的路徑用左 0 右 1 的方式加入 arr 陣列，這樣就完整建立 huffman tree code table
- 3 把輸入陣列判定出現頻率並計算 minimum weighted external path length 還有用上面提到的 queue 將根據 index 來把小於 rear 中存訪的陣列輸出

- 遇到問題與困境，與解決辦法:

◆ PROBLEM-2

1. 遇到困境(preorder 和 postorder 輸出):
2. 解決辦法:

建立二元搜尋樹的過程相對第一題簡單，只需要將 node 判斷大小存放左右 subtree，所以相對要面對的是建立樹之後的 preorder 跟 postorder，就是用上課學到的方式分別是中左右/左右中，來把前序後序寫出

- 程式內容:

- 1 第一步當然是建立結構跟節點，還有一些 inorder 的函式根加入二元搜尋樹的操作，來讓比較數字大小及 insert 函數可以用 linked-list 表達出來並成功建立二元搜尋樹
- 2 再來寫出兩個 function 分別印出 preorder / postorder，只是重複呼叫自身函數根據前序後序的規定次序來將左指標跟右指標依序指向