

## 資料結構與程式設計 程式作業二

**Input:** tree.txt

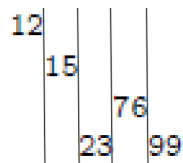
- (1) tree.txt 總共有 100 行，每一行代表一個 binary search tree。
- (2) 每一行有若干個相異二位數正整數，兩兩之間以一個空白分隔。
- (3) 每一行由左至右為 insert 到 binary search tree 的順序。
- (4) 助教會將 tree.txt 的路徑傳進你的程式，例如：python3 hw2.py ./tree.txt  
。請不要把路徑寫死在程式碼裡面。

**Output:** parenthesis\_representation.txt, textual\_printing.txt, left\_boundary.txt

- (1) 程式執行完之後需要生成三個檔案(但請不要寫成三支程式)。
- (2) parenthesis\_representation.txt 的每一行與 tree.txt 一一對應，總共 100 行。  
每一行請印出相應的 parenthesis representation。
- (3) left\_boundary.txt 的每一行與 tree.txt 一一對應，總共 100 行。每一行請印出相應的 left boundary。left boundary 請由 root 開始依照深度排序，兩兩數字間請用一個空白隔開。
- (4) textual\_printing.txt 請依照 tree.txt 中的順序將 binary tree 印出來。兩兩 binary tree 之間請用一個空行隔開。
- (5) 所有的 output files 請生成在跟你的 source code 同一個資料夾內。

**提醒：**

- (1) parenthesis representation 請不要印出多餘的空白。例如：  
19(15(13(12 14)17(16 -))23) 其中 17 前面和 23 前面是沒有空白的。
- (2) textual printing 每棵樹的深度就是它佔的行數，並請注意使用正確的“空白”進行排版(請不要用'\t')，數字彼此之間在直向不會重疊(如下圖)。



**繳交方式：**

- (1) 請將所有 source code (\*.py, \*.c, \*.cpp, \*.js, \*.java, ...) 放在以“學號”命名的資料夾內，如：b06901000。
- (2) 請寫一個 README.txt 檔案說明如何 compile、執行你的程式，以及任何你認為助教需要知道的資訊，例如：compiler 版本、python 版本、你使用的作業系統或執行的環境(windows, Ubuntu, Arch, ...)、使用到的套件等等。若助教在測試

時產生 error message 會先以你提供的資訊嘗試解決。

(3)請“不要”上傳任何除了 README.txt 和 source code 以外的檔案。(不要上傳 output files、不要上傳 tree.txt、不要上傳任何 compile 過的檔案和 compile 過程中出現的檔案)。

(4) 助教會根據你在 README.txt 中的說明去 compile、執行、得到你的 output files。請確保你上傳的所有檔案能夠根據 README.txt 的說明再現出你的結果。

(5) 請將以你學號命名的資料夾用 zip 壓縮上傳至 ceiba。

#### 批改方式：

(1) 助教會直接用程式去比對每個 output file 的結果(不會手動打開你的 output files 去檢查)。所以請不要印出任何多餘的東西，包括在不正確的地方印出空白、空行。可能會導致一行錯誤或是後半部份全錯。

(2)每一行的最末有多餘的空白是可以的。每一個 output file 的最末有多餘的空白或空行是可以的。

(3) 三個檔案各 100 個結果，一個結果一分，最後三個檔案的總分加起來除以三。

格式範例請參考附在資料夾內的檔案。

範例的 output files 是由 test\_case.txt 產生。

如有問題，請寄信至 [r06921038@ntu.edu.tw](mailto:r06921038@ntu.edu.tw)。