

## Demo 規定輸入、輸出

一、**Tree Traversal**：給定一棵樹，輸出 3 種序。

Preorder(NLR)：2 分

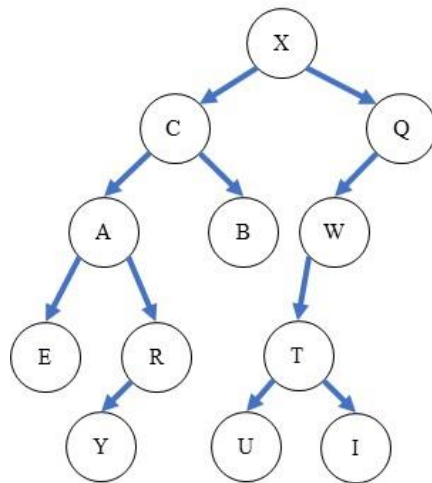
Inorder(LNR)：2 分

Postorder(LRN)：2 分

規定 Input：一串文字陣列，一個字元代表一個 Node，**空白**代表**無資料**；**最多輸入 63 個 Node**，資料有可能會重複。

規定 Output：分別輸出 Preorder、Inorder、Postorder 三種序，一種序 2 分。

範例 Input：以下圖之樹為例，輸入字串(因空白不易分辨先用\*代替空格，Demo 當天會使用空格)：XCQABW\*ER\*\*T\*\*\*\*\*Y\*\*\*\*\*UI\*\*\*\*\*



範例輸出：

Preorder(NLR)：XCAERYBQWTUI

Inorder(LNR)：EAYRCBXUTIWQ

Postorder(LRN)：EYRABCUIITWQX

加分：畫出樹者(Input 的樹)，**另加 3 分**。

Hint：利用遞迴(Recursion)的方式去做。不可以使用寫死的方式輸出，寫死的不予計分。

二、Tree 還原(6分)：給定 Preorder 及 Inorder 或 Inorder 及 Postorder，還原原本的樹。

規定 Input：兩串字串分別為 Preorder 及 Inorder 或 Inorder 及 Postorder，最長 63 個字元，資料不會重複。

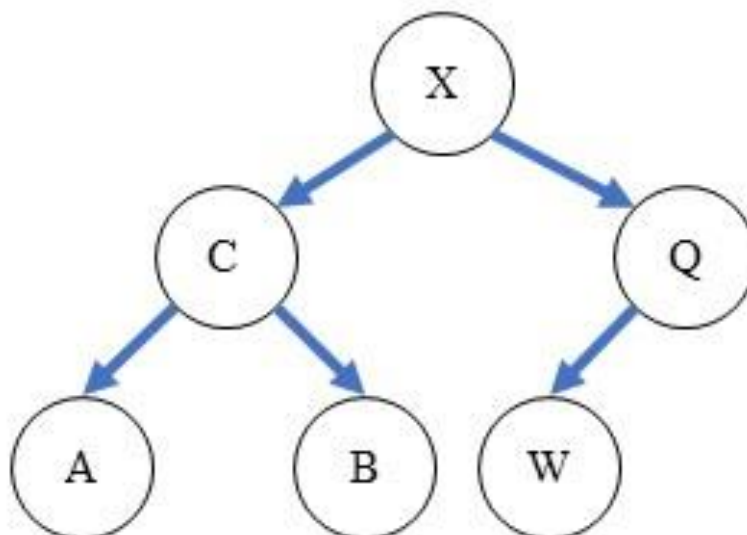
規定 Output：一顆還原後的樹。

範例 Input：

Preorder：XCABQW

Inorder：ACBXWQ

範例 Output：輸出如下圖

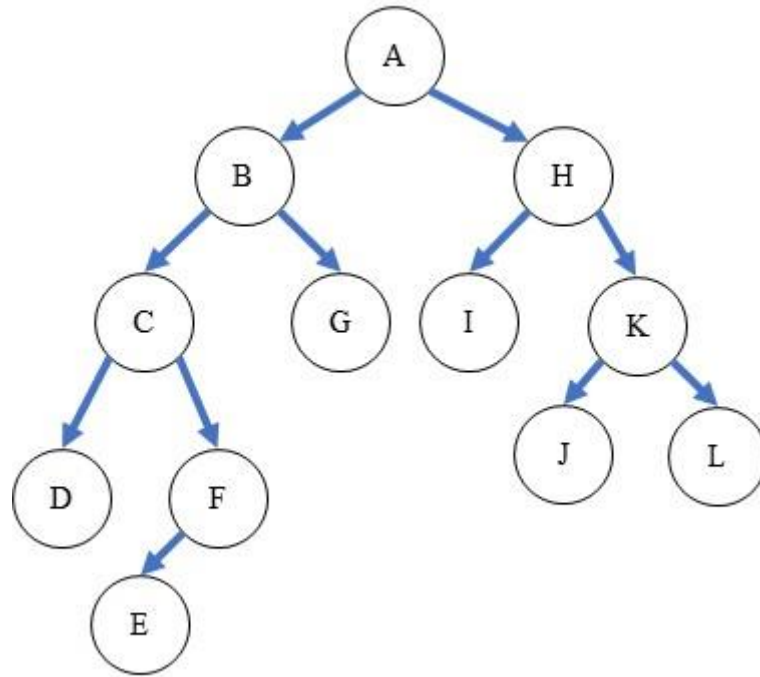


範例 Input2 :

Inorder : DCEFBGAIHJKL

Postorder : DEFCGBIJLKHA

範例 Output2 : 輸出如下圖



Hint : 可先決定樹高再決定每個 Node 的位置。因本題的輸入限制，因此樹高最高六層。

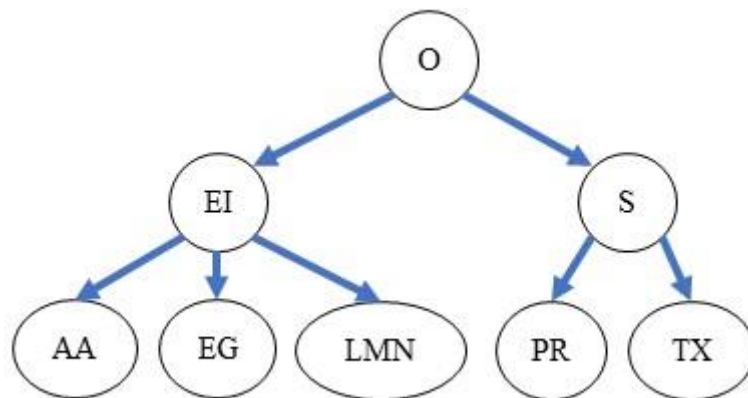
可以以純文字輸出一個樹(輸出方式同第一題的輸入)，但本題分數剩下 3 分。

三、2-3-4 Tree 轉 Red-Black Tree(8分)：輸入一串 Data 為一棵 2-3-4 Tree。請轉換成 Red-Black Tree。

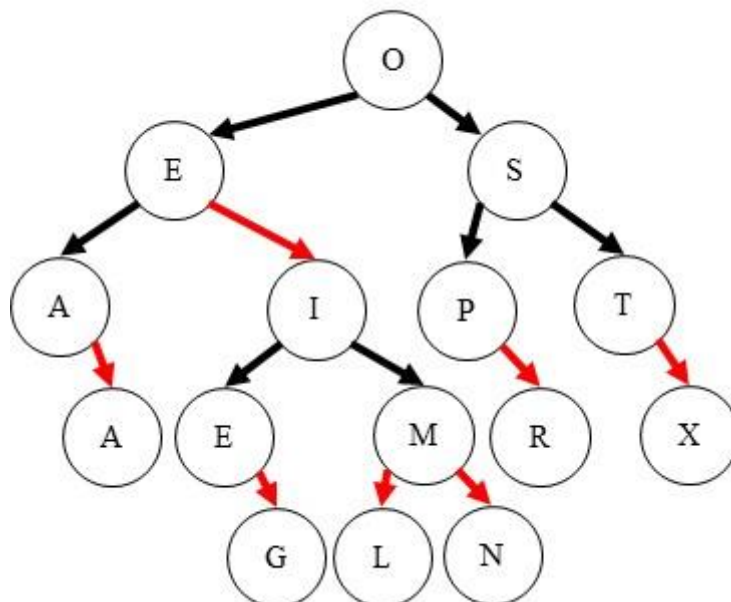
規定 Input：一串字串代表一棵 2-3-4 樹，每個 Node 使用「,」隔開，因為 2-3-4 樹最多可以有 4 個 Node，因此空的 node 使用\*代替。最多 21 個 Node。

規定 Output：一顆畫好的 Red-Black Tree

範例 Input：O,EI,S,\*,\*,AA,EG,LMN,\*,PR,TX,\*,\*,



範例 Output：輸出如下圖



Hint：一定要畫出圖。

四、上機考(加分題)：題目會在當天公布，共 5 分。