➢ Binary Tree

二元樹對電腦歌學來說是一個非常常用的資料結構。一株二元樹基本上由

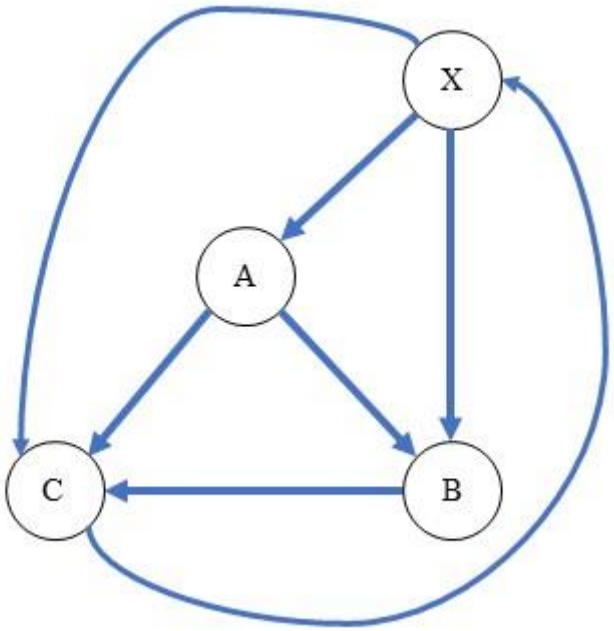
$\begin{cases} node\ 節點 \\ edge\ 邊 \end{cases}$ 所構成。

在二元樹中，每一個節點最多只會有兩條「指出」的 edge。

另外在諸節點中，有一個特別的 node 稱為 root（根）位在一株 tree 中的最上層，只有 root 是沒有「指入」的邊的，樹中其他節點接洽有一條指入的邊。

在 tree or graph 這類類的圖形結構中，當它具有方向性時，我們會定義頂點的 Indrgree 唯一節點指入的邊數，Outdegree 唯一節點指出的邊數。

例如：



在這 graph 中 Indegree(A)=1

Outdrgree(A)=2

Indegree(B)=2

Outdrgree(B)=1

Indegree(C)=3

Outdrgree(C)=1

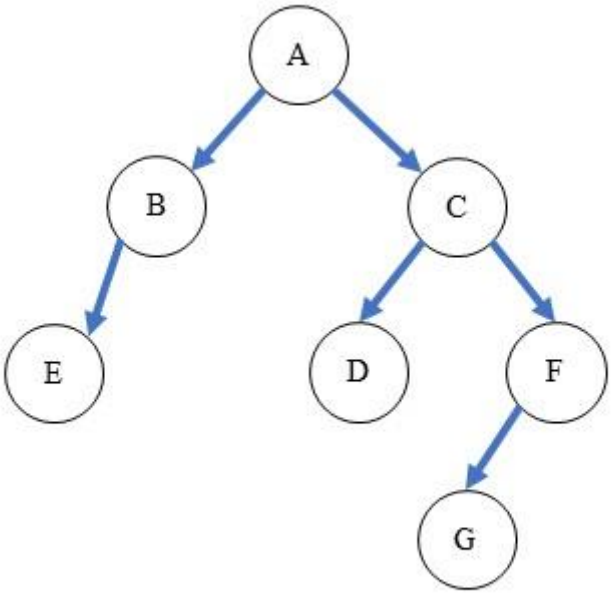
Indegree(X)=1

Outdrgree(X)=3

➢ Tree 的定義

一株 T，是有限個節點所構成的集合，對於T－{R}後所剩下的其他節點，可以分成若干個互斥的子集合，又稱為 Subtree(子樹)。

例如：



Ⓐ為 root

Ⓑ-Ⓔ為 left subtree

Ⓒ-Ⓓ-Ⓕ-Ⓖ為 right subtree

左右子樹間互斥，如果刪去 A，則左右子樹成為兩株互斥獨立的新樹，而

這樣由複數株樹所構成的 data structure 稱為 forest。

樹狀結構中的節點，是只有階層關係的，被一條 edge 連結起來的兩節點間，$\begin{cases} parent\ node\ 父節點 \\ (ancestor\ node)：在上者 \\ son\ node, child\ node\ 子節點 \\ (decendent\ node)：在下者 \end{cases}$。

如上圖，Ⓐ是ⒷⒸ的 parent node，是ⒹⒼ的 ancestor node；而Ⓔ是Ⓑ的 child node，Ⓖ是ⒶⒸⒻ的 decendent node。

另外，屬於同一個 parent 的 child nodes 之間，有 sibling 的關係（或稱為 brother node）。

在 tree 中，底下沒有 child node 的節點被稱為，

ⒺⒹⒼ$\begin{cases} leaf\ node\ 葉節點 \\ treminal\ node\ 終端(節點) \\ ectrenal\ node\ 外部節點 \end{cases}$ 相對地，具有至少一個 child node 的的節點稱回 internal node 內部節點。

一個節點上所連結的 edge 數(branch 數)被稱為該節點的 degree 又可分為 Indegree, Outdegree。

由某個節點，假設為 I，到達另一個節點，假設為 J 之間所必須經過(拜訪，visit)的節點，構成 I→J 間的 path，而 part 上的 edge 數，稱為 length of path(路徑長度)。

例如：

Ⓐ到Ⓖ的路徑為Ⓐ→Ⓒ→Ⓕ→Ⓖ length of path 為 3

- **level(階度)**

以 root 的 level 為 1，以下每往下一層(與 root 的路徑長度遞增 1)，則同層個節點的 level 均遞增 1。

➢ height(高度)

一株 tree 的最大高度稱為 height(樹高)。如上圖樹高為 4。

height 又稱為 depth。

➢ Tree Traversal(二元樹追蹤)

在拜訪二元樹中的節點時，如果沒有依照，一定順序的話，就很容易錯過某些 branch 沒有拜訪到，為了要確保所有節點都會被經過，所以我們需要某種順序，去追蹤 tree 中的 nodes，這時，依順序的不同可分為：

$$\begin{cases} NLR(Preorder) \\ LNR(Inorder) \\ LRN(Postorder) \\ NRL(converse\ Preorder) \\ RNL(converse\ Inorder) \\ RLN(converse\ Postorder) \end{cases} \quad \begin{cases} N為父節點 \\ L為左節點 \\ R為右節點 \end{cases}$$

考試可能出給你 Preorder 追蹤的結果：ABECDFG

及 Inorder 的結果：EBADCGF

則 Postorder 為：EBDGFCA