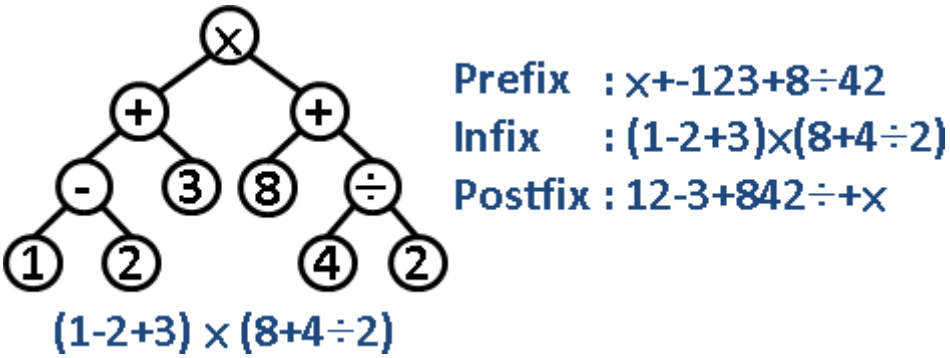
二元樹的建立主要是因應其儲存之資料，例如運算樹：



最簡單的方式是輸入後序運算式，放入堆疊中，需要時從堆疊 pop 出來建立。

例如此運算樹：當讀入 1、2 時將運算元 push 進堆疊中，遇到運算子”-“時則從堆疊中 pop 出兩個運算元建成樹，之後再將節點”-”放入堆疊中，重複此動作直到運算式讀取完成即可形成運算樹。

如果想要建立任意形狀的二元樹的話，推薦的是使用佇列建樹，或是階序建樹，也稱為 BFS（廣度優先搜尋）方式建樹。

此種方式的建立原理是，先從佇列中讀取一個節點，輸入該節點的左子樹右子樹，再將其放入佇列。此方法會從樹的根節點開始往下建立子樹，若左子樹不為空時放入佇列，再輸入右子樹，再放入佇列，從上到下、從左到右依序輸入所有節點，因此可形成任意形狀二元樹。

例如：輸入 1 1 1 2（第一個參數代表左子樹是否為空，為空輸入 0，否則 1，第二個參數代表左子樹節點的值，第三個參數則為右子樹是否為空，第四個參數為右子樹節點的值）產生以下的樹：

0（此節點為事先建立的）

1 2

此時佇列中依序有 1、2 這兩個節點，接著可以輸入節點 1 的子樹，例如輸入：1 3 1 4，則樹變為：

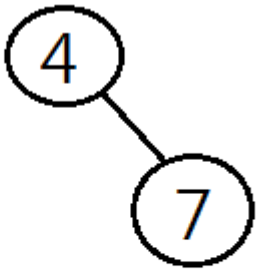
0

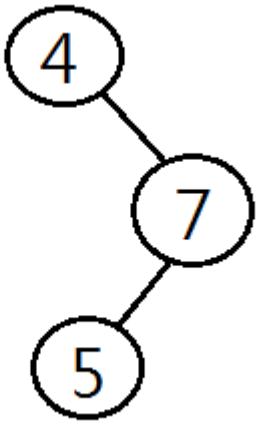
1 2

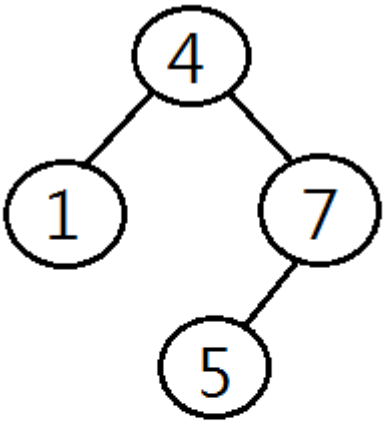
3 4

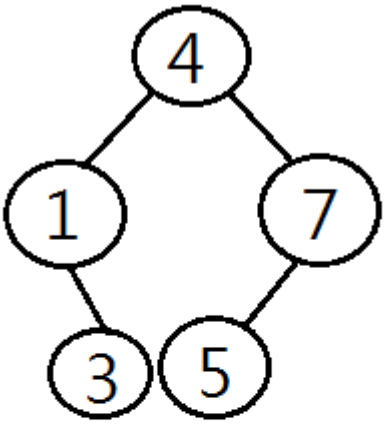
而此時佇列中有 2、3、4 三個節點，依此方法循序下去可建立出任意形狀的二元樹。

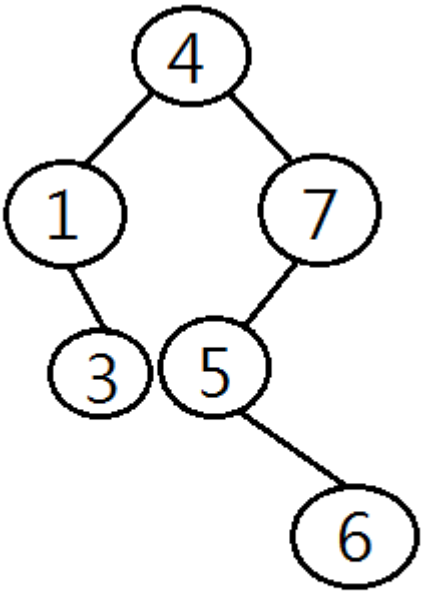
另外還有一種樹叫做二元搜尋樹，二元搜尋樹簡單來說就是用二分法將數字分配左子樹或右子樹，此種方式為最易建出樹的方式。若數字小於樹根，則擺在左子樹，反之擺在右子樹，例如底下輸入：4 7 5 1 3 6 8

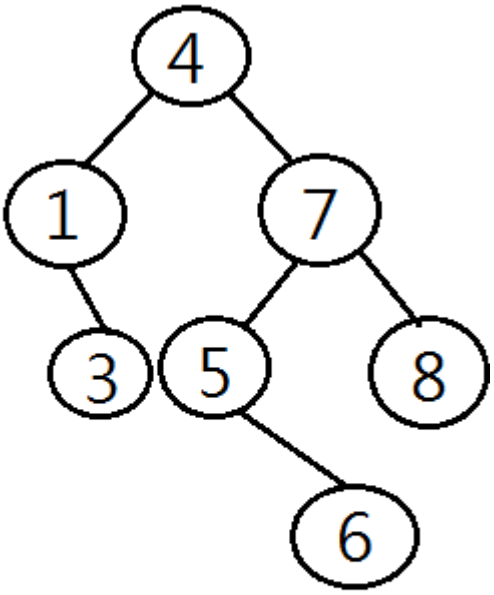





4
1
7
5


4
1
7
3
5





此樹根據中序走訪結果為：1345678