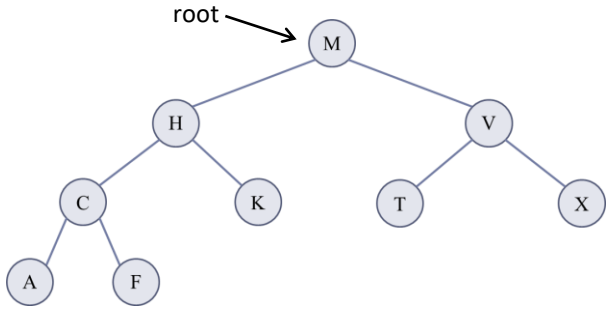
Reference-based Representation

What does a Binary Tree node look like?

Using what we know from linked-lists, how can we traverse a tree?

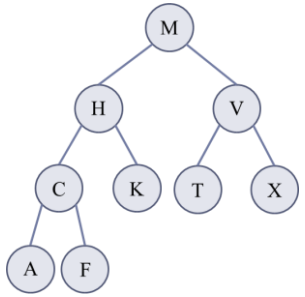


Observation:

How can we add a node to an existing tree at a certain location?

Array-based representation

What would an array look like representing the same Binary Tree?

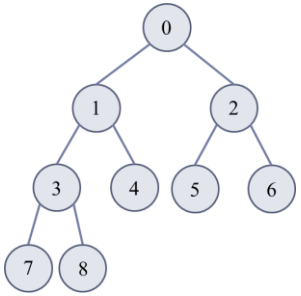


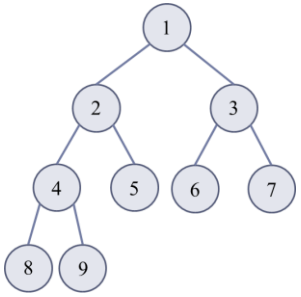
Two methods:

M	H	V	C	K	T	X	A	F						
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

-	M	H	V	C	K	T	X	A	F					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Viewing the tree by the index number of each item:





Left child:
Right child:
Parent:

Left child:
Right child:
Parent:

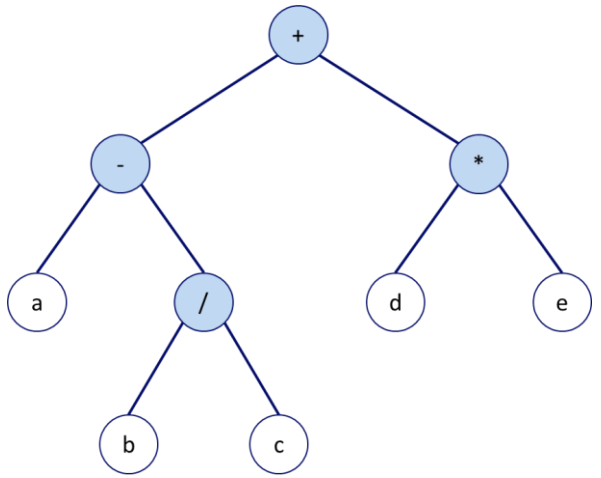
What index would we insert the value **U** so that it was **T**'s right child?

Exercise: Write a method to calculate the height of a tree

```
public int height(TreeNode n) {



}
```



Traversals

In-Order:
Pre-Order:
Post-Order
Level-Order

Evaluating the tree expression:

1. Traverse through each item in the post-order expression

2. If item is an operand, push to stack. Otherwise, pop two elements.
   i. Let A be first popped element
   ii. Let B be second popped element
   iii. Evaluate B <operator> A
   iv. Push result to stack

3. Pop final item and return it.