

CS 260

Trees on Arrays

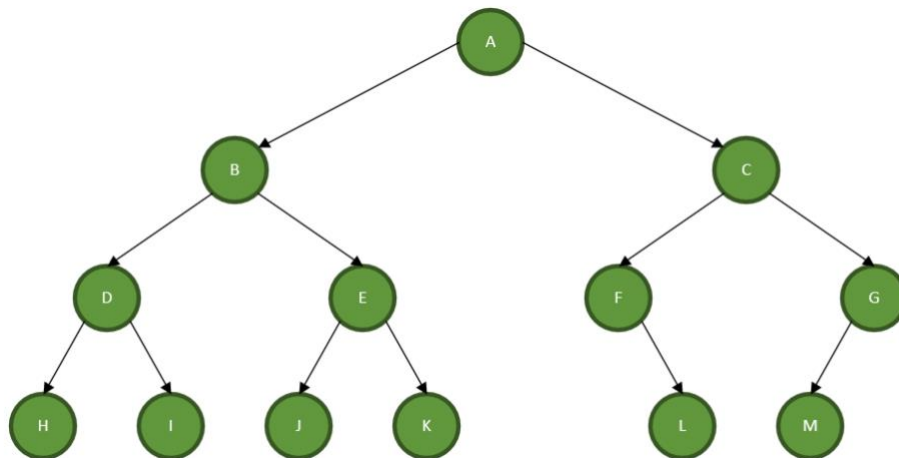
Overview

Previously, binary trees have been implemented in a manner like linked lists, that is each node contains references leading to other nodes (parents, left child, right child). The other way to implement a binary tree is on an array where the index is used to compute the location of the parent and children.

Binary Trees on Arrays

When implementing a binary tree on an array, the root is typically located at 1. Then its left child is at index 2 and the right child at index 3. In general, if a node is at index i , the parent is at index $i/2$, the left child at $2i$ and the right child at $2i+1$.

Consider the following example:



The array that implements this is:

Value	A	B	C	D	E	F	G	H	I	J	K		L	M	
Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Here the root A is at index 1, its right child C is at index $3 = 2 \cdot 1 + 1$. C's left child F is at index $6 = 2 \cdot 3$ and the right child G is at index $7 = 2 \cdot 3 + 1$. The parent of M (index 14) is G and is at index $7 = 14 / 2$.

Considerations

Since the tree starts at 1 and builds down from there, increasing the array size is a simple matter of creating a new larger array and copying items straight across index to index. Notice that the number of nodes at each level is one more than the total number of nodes in all previous levels, so when you need to increase the array by doubling it only allows one more level of nodes.