

Graph Traversal



Prof John Williams

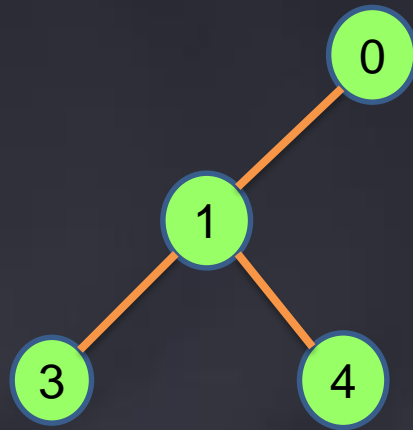
Dr. Abel Sanchez



Trees- Basic Data Structure

A node may have at most one parent but many children. Eg node 1 has parent 0 and children [3, 4]. Thus, the minimum data structure for any node is as below

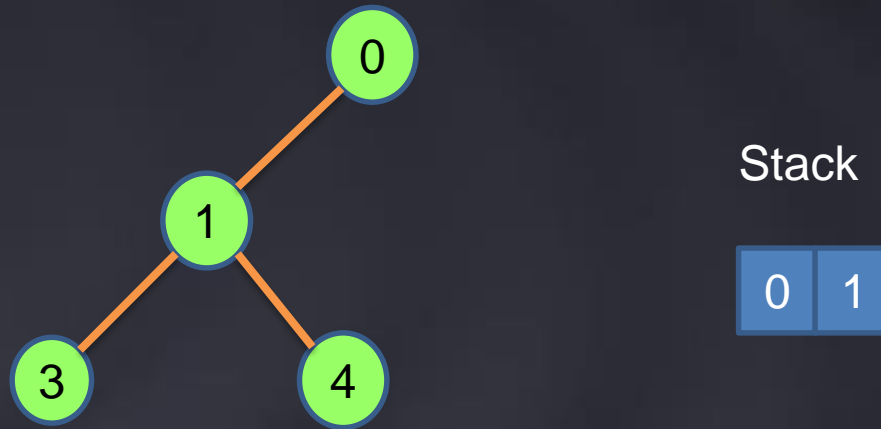
node = {children : children [] }



Basic Data Structure – Leaving a Trail

`node = { children : children [] }`

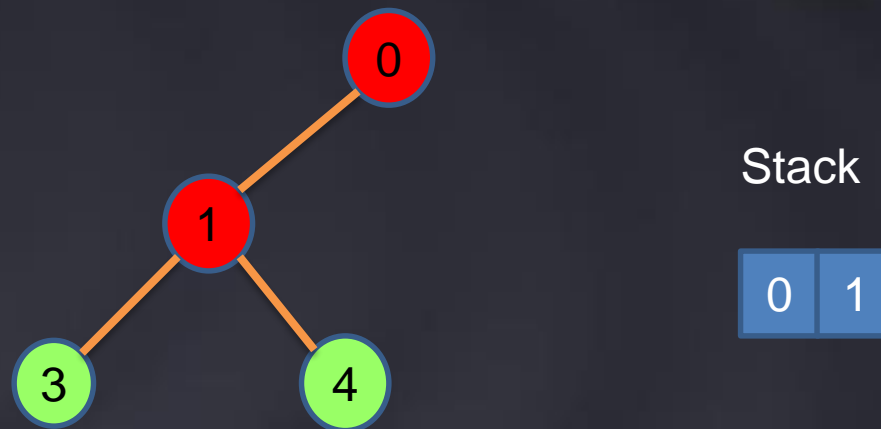
If we want to keep track of a node we store a reference to it in an array. Eg [0, 1] might indicate we have moved from node 0 to 1. Sometimes these are called 'bread crumbs' because they allow us to retrace our steps.



Basic Data Structure – Flagging nodes visited

node = {children : children [], color:'Red' }

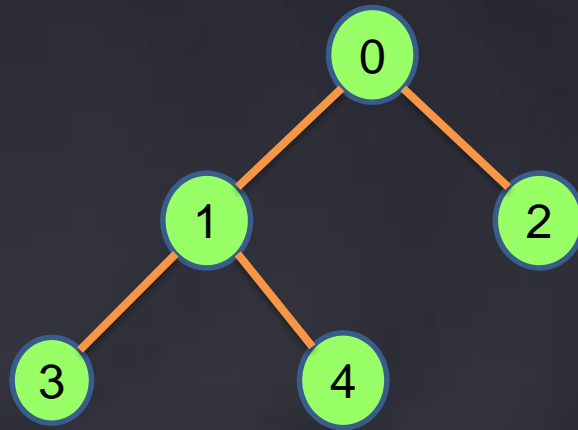
Another mechanism we use is to color nodes we have 'visited' so that we don't repeat our steps. Here I'm going to color nodes I've visited 'Red'



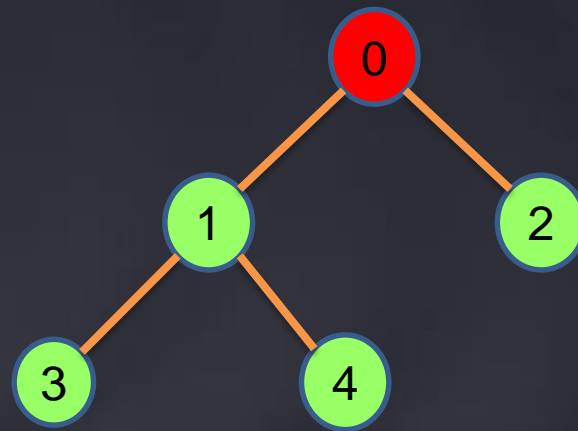
Now we are ready to systematically explore our tree using the array to retrace our steps and the color to flag that we've already explored part of the tree. Depending on the way we explore the tree we use the array as a Stack or a Queue.

Depth First Search

Stack



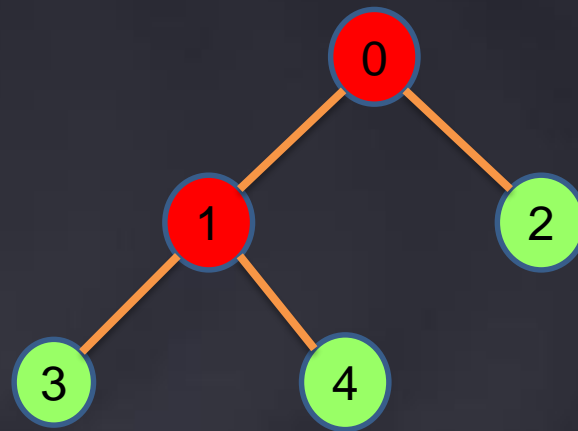
Depth First Search



Stack

0

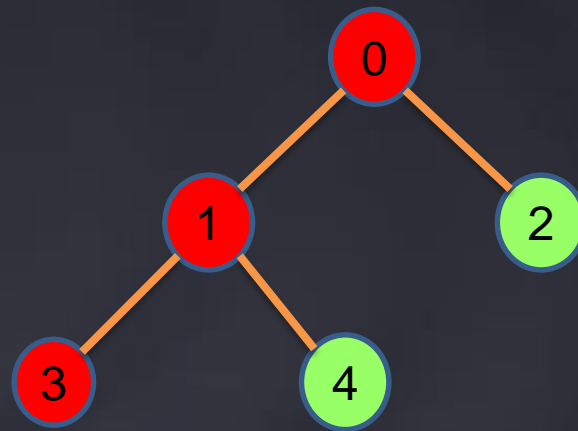
Depth First Search



Stack



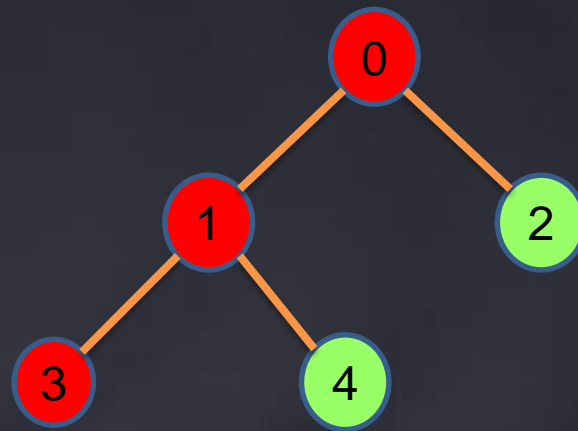
Depth First Search



Stack



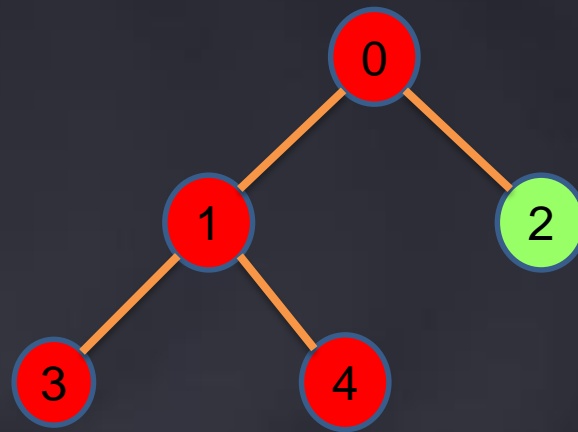
Depth First Search



Stack



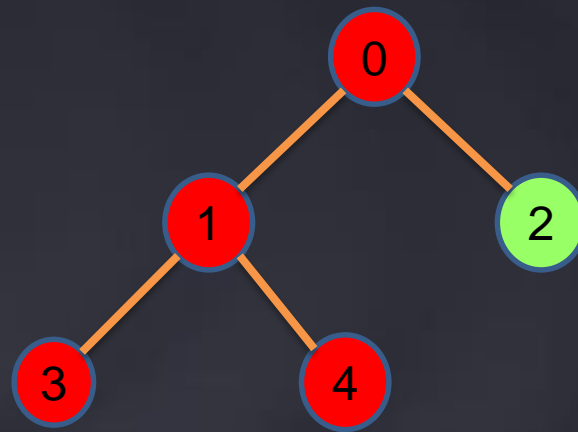
Depth First Search



Stack



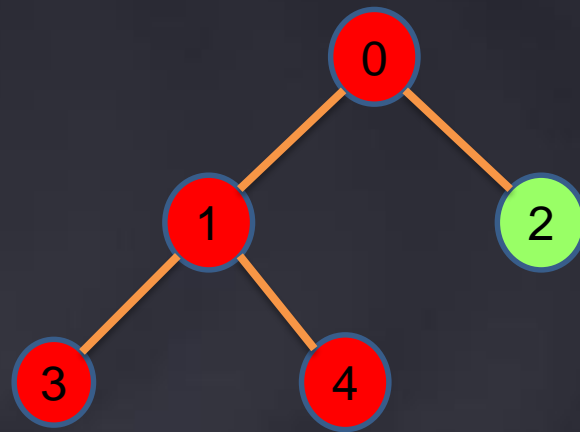
Depth First Search



Stack



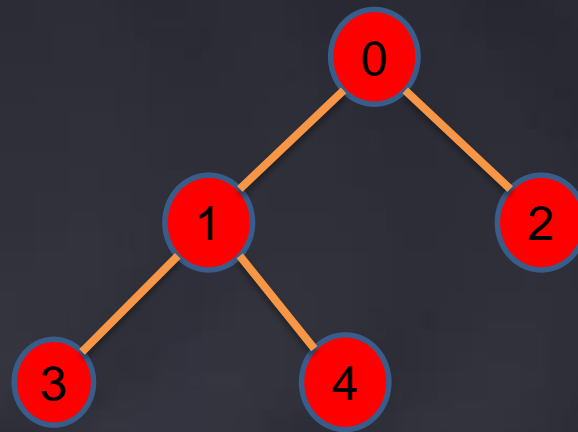
Depth First Search



Stack

0

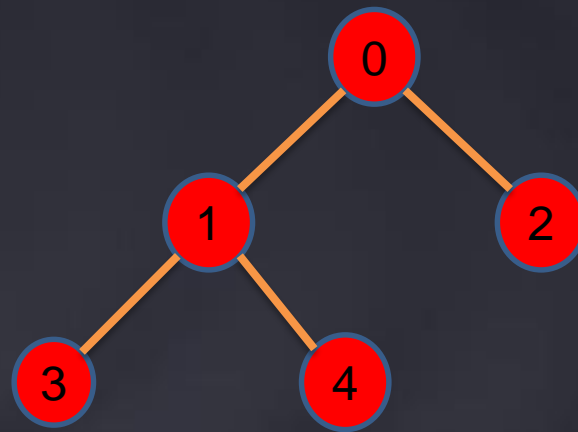
Depth First Search



Stack



Depth First Search

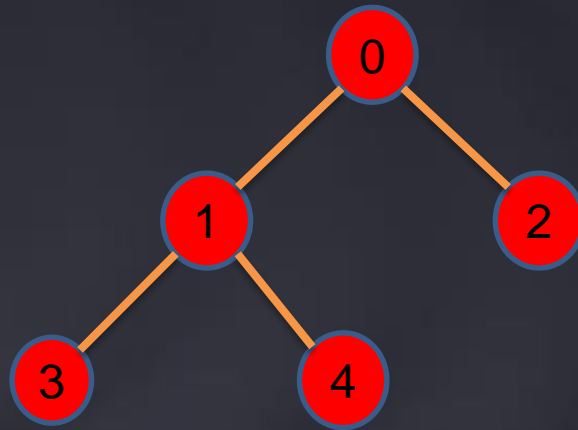


Stack

0

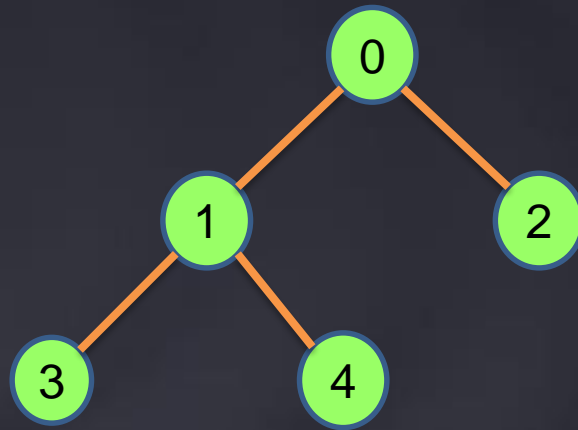
Depth First Search

Stack

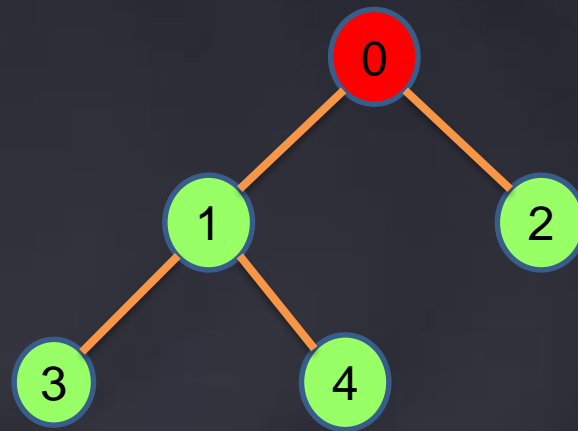


Breadth First Search

Queue



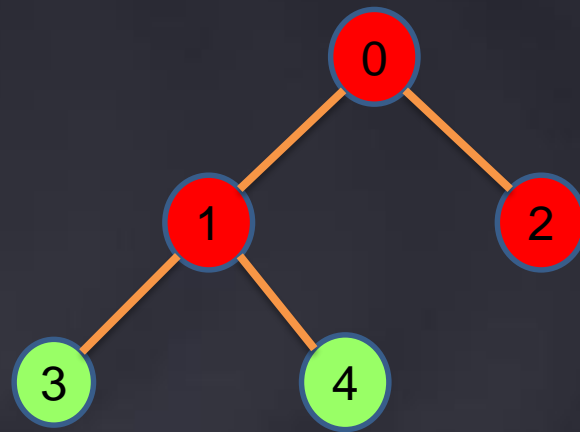
Breadth First Search



Queue

1

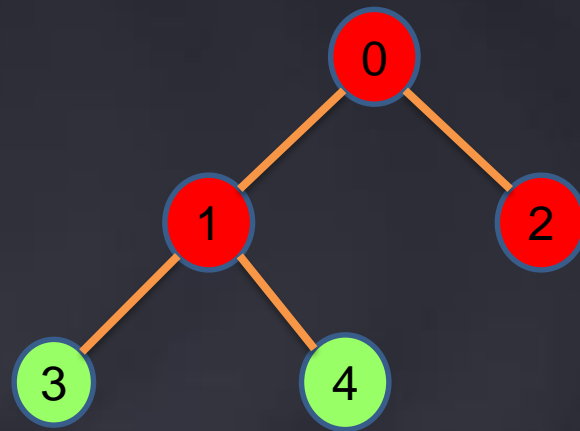
Breadth First Search



Queue



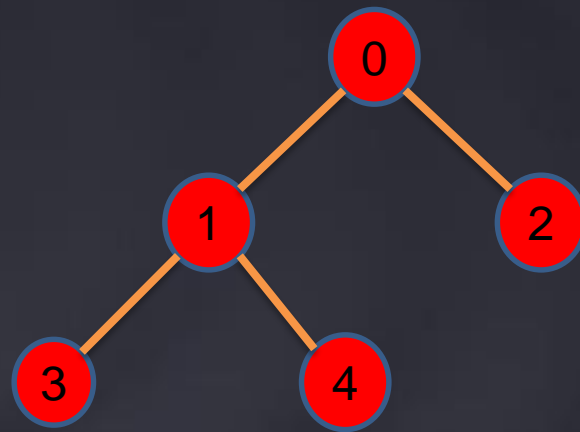
Breadth First Search



Queue

2

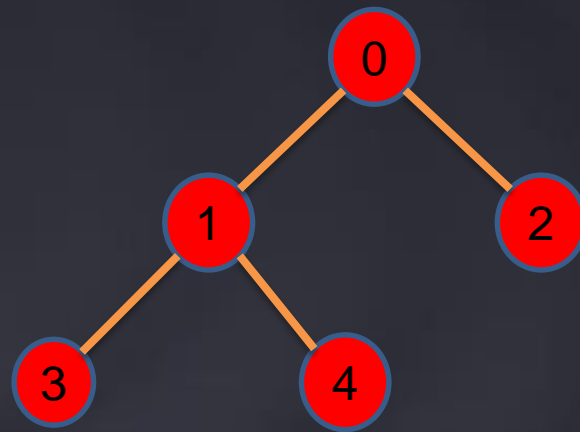
Breadth First Search



Queue



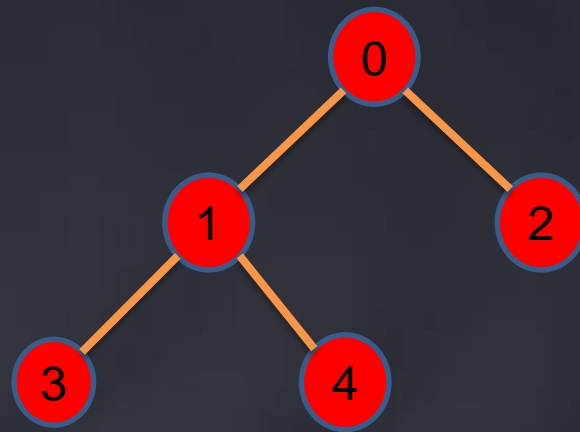
Breadth First Search



Queue



Breadth First Search



Queue

4

Breadth First Search

Queue

