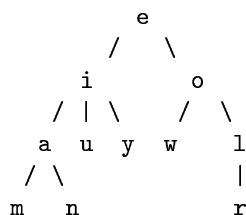


Lecture 36, part 1: Stacks, queues, and search

Depth first and breadth first search in a tree can be described using the same rule, differing in whether the rule uses a stack or a queue. Consider the following tree, root at the top.



The following rule will print out the labels of the nodes of this tree in depth first order. Initialize a stack containing just the root of the tree, e. While the stack is nonempty, pop the top element, print it out, and push the children of the top element onto the stack (in right to left order, so that they will be processed in left to right order.) The successive states of the stack are as follows (top of stack at left):

```

stack: e      pop and print e, push children o and i
stack: i o    pop and print i, push children y, u and a
stack: a u y o pop and print a, push children n and m
stack: m n u y o pop and print m, no children to push
stack: n u y o pop and print n, no children to push
stack: u y o   pop and print u, no children to push
stack: y o    pop and print y, no children to push
stack: o      pop and print o, push children l and w
stack: w l    pop and print w, no children to push
stack: l      pop and print l, push child r
stack: r      pop and print r, no children to push
stack: empty  printing done

```

To print out the labels of the nodes of the tree in breadth first order, all we need to do is modify the rule above to use a queue instead of a stack and add the children of the printed node to the end of the queue (in left to right order). With this change, the successive states of the queue are as follows (beginning of queue at left):

```

queue: e      dequeue and print e, enqueue children i and o
queue: i o    dequeue and print i, enqueue children a, u and y
queue: o a u y dequeue and print o, enqueue children w and l
queue: a u y w l dequeue and print a, enqueue children m and n
queue: u y w l m n dequeue and print u, no children to enqueue
queue: y w l m n dequeue and print y, no children to enqueue
queue: w l m n dequeue and print w, no children to enqueue
queue: l m n  dequeue and print l, enqueue child r
queue: m n r  dequeue and print m, no children to enqueue
queue: n r    dequeue and print n, no children to enqueue
queue: r      dequeue and print r, no children to enqueue
queue: empty  printing done

```