Trees

George Tang

5 October 2018

1 Definitions

A tree is a special type of graph that satisfies two constraints:

- All nodes must be connected by edges.
- There are no *cycles*, a set of distinct edges that can be followed from any node to reach itself.

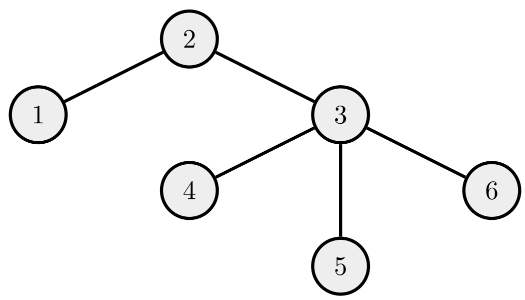

Figure 1: This diagram illustrates a tree.

Every node c except the *root* is connected to some other node p called the *parent* of c. We also call c the *child* of p. Nodes that have no children are called *leaves*. The *ancestors* of a node are all the nodes on the path between it and the root.

2 Implementation

There are two ways to implement a tree data structure:

- Nodes are defined as a class with a list of children, pointer to parent, and the data it holds.
- Node information are stored in many arrays (parent, children, data), where the index corresponds to the node id. For the children array, each index holds a linked list of children node ids. This one is the favorite of competitive programmers.

3 Traversals

There are three types of tree traversals. Each traversal is recursive, meaning you repeat the traversal until you reach a leaf node or all children have been visited

- *Preorder*: visit the current node first, then the children nodes.
- *Postorder*: visit all children nodes then the current node.
- *Inorder*: when there are only two children, and the child with a smaller data (or equal to) value than the current node's data value is the left child, and the child with the larger value is the right child. Visit the left node, then then current node, and then the right node.

4 Binary Search Trees

A binary tree has only two children. In the inorder traversal section, we introduced a binary search tree. BSTs are often very useful in contest, especially when there requires a $logN$ search query on an enumerable data set. However this requires the BST to be *balanced*, meaning that the left and right subtrees of the current node are not too different in size. This occurs when the data is inserted in random order. Once we cover insertion, convince yourself this is tree. For the following implementations, we assume the array representation (left refers to child[0] and right child[1]). The tree representation can be easily extended from the following.

4.1 Search

Algorithm 1 BST Search

```
function SEARCH(curr, u)
    if curr.val = u then
        return curr
    if curr.left = null and curr.right = null then
        return null
    if curr.val < u then
        return SEARCH(curr.left, u)
    else
        return SEARCH(curr.right, u)
```

4.2 Insert

Algorithm 2 BST Insert

```
function INSERT(curr, u)
    if curr.val =< u then
        if curr.left = null then
            curr.left = Node(u)
        else
            INSERT(curr.left, u)
    else
        if curr.right = null then
            curr.right = Node(u)
        else
            INSERT(curr.right, u)
```

4.3 Delete

Three cases for node to be deleted:

- no children
- one child
- two children

Algorithm 3 BST Delete

```
function MINNODE(curr)
    while curr.left != null do
        curr = curr.left
    return curr
function DELETE(node, u)
    if node.left = null and node.right = null then
        node = null
    if node.left! = null and node.right! = null then
        temp = MINNODE(node)
        node.val = temp.val
        temp = null
    else
        if node.parent.left = node then
            if node.left = null then
                node.parent.left = node.right
            else
                node.parent.left = node.left
        else
            if node.left = null then
                node.parent.right = node.right
            else
                node.parent.right = node.left
```

4.4 Lazy Delete

If memory is not an issue, you can just keep a flag that indicates whether the node has been deleted or not.

5 Problems

5.1 USACO Gold February 2018, Cow at Large

The farm consists of ($2 \leq N \leq 10^5$) barns and $N - 1$ bidirectional tunnels. Every barn is connected. Every barn with one tunnel is an exit. Bessie will be at any barn and attempt to exit. The moment Bessie starts walking, farmers will start at various exit barns and attempt to catch Bessie. The farmers and Bessie can traverse one edge per unit time. Everyone knows where everyone else is at all times. The farmers catch Bessie if at any instant a farmer is in the same barn as Bessie, or crossing the same tunnel as Bessie. Given that Bessie starts at barn K, help Bessie determine the minimum number of farmers needed to catch Bessie.