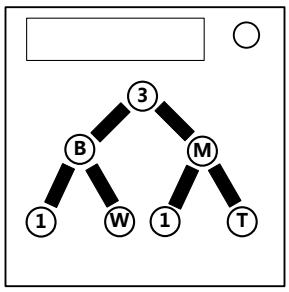
On the Subject of the Binary Tree



Contrary to normal trees, binary trees grow downward and are endlessly confusing with their different traversal orderings...

See Appendix BT for a glossary of terms and orderings relating to binary trees.

- The module contains a screen and a depth-three complete binary tree, containing seven nodes that are buttons which could be pressed. Each node may be differently colored, and each has a single black or silver character printed on it.
- The module contains three stages. In each stage, determine which node to press using a reference node. For the first stage, use the root of the tree as the reference node. For each subsequent stage, use the correct node pressed in the previous stage as the reference node.
- To find which node to press, first turn the character printed on the reference node into a number. If it is a numeric character, use that number, otherwise use the position of that character in the alphabet + 2.
- If the resulting number is greater than 7, subtract 7 repeatedly until it is within the range 1~7. If the number is 0, use 7. Call this number **N**.
- Use the table on the next page to determine an ordering of the nodes from the current stage number and the color of the reference node.
- Press the node that is the **N**th node from the start of the ordering, **UNLESS** the text on the reference node is silver, in that case press the **N**th node counting in reverse from the end of the ordering instead.
- The screen displays previously pressed correct nodes, and is not required to solve the module.
- Pressing a wrong node will result in a strike, but the module will stay on its current stage. The module will also display the cross-shaped strike indicator for a while, during which all inputs are ignored.

Node Orderings Table

Stage #:		Stage 1	Stage 2	Stage 3
Ref Node Color:	Red	Preorder	Right-to-Left Inorder	Right-to-Left Level Order
	Green	Inorder	Right-to-Left Postorder	Inorder
	Blue	Postorder	Right-to-Left Preorder	Right-to-Left Inorder
	Orange	Level Order	Level Order	Preorder
	Cyan	Right-to-Left Preorder	Preorder	Level Order
	Magenta	Right-to-Left Inorder	Inorder	Right-to-Left Postorder
	Yellow	Right-to-Left Postorder	Right-to-Left Level Order	Postorder
	Gray	Right-to-Left Level Order	Postorder	Right-to-Left Preorder

Appendix BT: Binary Tree Terms & Orderings

Terms:

- A **complete binary tree** is a structure in which each node, except for the nodes at the bottom layer, is directly connected to two other nodes under it, called its **children.**
- The **root** of the tree is the node at the top layer of the tree.
- The **left subtree** of a node is the tree containing the node's left child, and all of the child's children. The **right subtree** of a node is the tree containing the node's right child, and all of the child's children.
- An **ordering** of the tree visits each node in the tree exactly once in a certain order.

Orderings:

- **Preorder:** Starting at the root, first visit the root, then visit all nodes in its left subtree in preorder, then visit all nodes in its right subtree in preorder.
- **Inorder:** Starting at the root, first visit all nodes in its left subtree using inorder, then visit the root, then visit all nodes in its right subtree using inorder.
- **Postorder:** Starting at the root, first visit all nodes in its left subtree in postorder, then visit all nodes in its right subtree in postorder, then visit the root.
- **Level Order:** Like reading order. Start at the top row, then read all nodes from left to right on each row.
- For the **Right-to-Left** version of an ordering given above, read the corresponding description above, but swap all occurrences of the word "left" and "right".