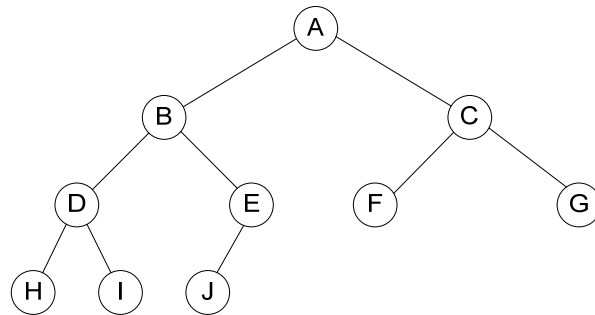


Lab 1 (Due Jan 25th)

Please make a complete tree—one that is completely full except possibly on the right end of the bottom row. The characters should be ordered from the top down and from left to right along each row, as if writing a letter on a pyramid. (This arrangement does not correspond to any of the three traversals we discussed in this chapter.) Thus, the string ABCDEFGHIJ would be arranged as



One way to create this tree is from the top down, rather than the bottom up. Start by creating a node which will be the root of the final tree. If you think of the nodes as being numbered in the same order the letters are arranged, with 1 at the root, then any node numbered n has a left child numbered $2 \times n$ and a right child numbered $2 \times n + 1$. You need to use a recursive routine that makes two children and then calls itself for each child. The nodes don't need to be created in the same order they are arranged on the tree.

An incomplete solution is provided in the `TreeApp.java` program. Please finish this program by completing the `public void recBuild(Node thisNode, int ourNumber)` function. *[Hint: you need to add codes from line 49]*