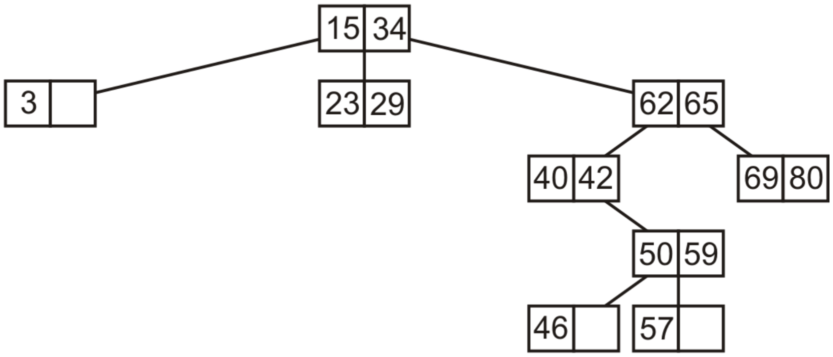
ECE 250 *Algorithms and Data Structure*
Department of Electrical and Computer Engineering
University of Waterloo

Please send any comments or criticisms to dwharder@alumni.uwaterloo.ca
with the subject ECE 250 Questions 6.4.
Assistances and comments will be acknowledged.

6.4*a* As you can see, a multiway tree can still turn into something that essentially looks like a linked list:



6.4*c* Assume that a node may not yet be full, but still contain sub-trees. Insert should always try to fill a node before inserting a value into a sub-tree, so as to not unnecessarily increase the height of the tree. This can be done as previously described in class, but now the sub-trees may have to be shifted, as well.

Erase could be implemented as follows:

1. If the sub-tree to the immediate right of the value is not empty, copy the minimum element from it into the current node and recursively call erase on that copied value in that sub-tree.
2. If the sub-tree to the immediate right is empty, but the sub-tree to the immediate left is not empty, copy the maximum element from it into the current node and recursively call erase on that copied value in that sub-tree.
3. Otherwise, just shift the elements and sub-trees stored to the right of the location of the element being erased over to the left by one.

6.4*d* $8^{6+1} - 1 = 2\,097\,151$

Multiply the depth by the number of nodes at each depth:

$$\frac{\sum_{k=0}^{6} 7 \cdot k \cdot 8^k}{\sum_{k=0}^{6} 7 \cdot 8^k} = \frac{12283320}{2097151} \approx 5.857 .$$

6.4*e* Use the same ideas from Question Set 6.3 *In-order traversals*.