Math 188, Winter 2001 March 7, 2001
Prof. Tesler

Homework #9, Due March 14

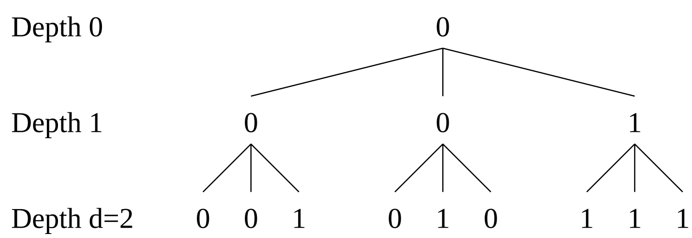
Chapter 7# 43 (*Hint:* how many digits are required if the numbers are in base r?)
and the problems below: H-7, H-8.

Problem H-7. Let T be the tree shown on page 327, Figure 8.7(a) (upper left corner). This is a B-tree where each node has a minimum capacity of 1 element and a maximum capacity of 2 elements.

(a) Show how to insert 6 into T. Then insert 9.5 into the result.

(b) Start over with T. Show how to insert 9.5 into T and then insert 6 into the result.

Problem H-8. Consider a tree of depth d in which all internal nodes have three children, and all leaves are on level d. At every leaf, we assign a value 0 or 1. At every internal node, the value 0 or 1 is computed by taking the majority of the values at its the three children (if two or three children have value x, the node is assigned x too).



You want to find the value assigned to the root. An upper bound on the number of leaves that you have to examine to compute this is 3^d (that is, all of them). However, if you are lucky, you may be able to examine fewer. For the leftmost group of three leaves, after you examine the first two, you already know the parent will be 0, so there's no need to examine the third.

Number the leaves from 1 to 3^d (going left to right in a picture like the one above).

The Standard Strategy: Examine the leaves in the order 1 to 3^d, skipping any leaf that is unnecessary because prior leaves already imply the value assigned to an ancestor of this leaf. In the tree shown above, we would examine leaf 1 (0) and then leaf 2 (0); this forces their parent to be 0, so we skip leaf 3. Then we would examine leaf 4 (0), leaf 5 (1), leaf 6 (0), and the parent of these three would be assigned the value 0. Now two children of the root are known to have value 0, so the root is forced to be 0 without even examining leaves 7, 8, 9. This strategy only examined 5 leaves.

(a) For a tree of depth d, if an algorithm were extremely lucky in its choice of the order it examined leaves, what is the absolute minimum number of leaves it would have to examine? (In the tree shown above, if we happened to examine leaves 1, 2, 4, 6, and no others, that would be sufficient to conclude the root has value 0.)

(b) Over all 2^{3^d} settings of the leaves in trees of depth d, what is the average number of leaves $A(d)$ that the Standard Strategy given above will examine?

(c) Prove that no matter what strategy is used for choosing the order in which to examine the leaves, there will be a tree that requires examining all 3^d leaves. Do this through an adversary argument. In other words, if an adversary were presented with a sequence of requests for the values at various leaves, and the adversary made up answers at the time the questions were asked (but kept the answers consistent with previous answers) rather than assigning values to the leaves at the beginning, how could the adversary systematically choose the answers to force us to request the values of all of the leaves before we knew the value of the root? (See Chapter 8.5 for adversary arguments.)