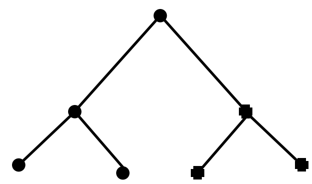
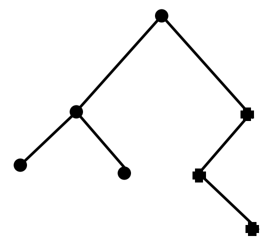
ECE 130 - TREES - INVESTIGATION 20
BINARY SEARCH AND DECISION TREES

WINTER 2004 **A.P. FELZER**

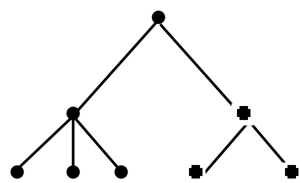
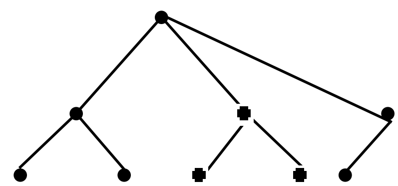
To do "well" on this investigation you must not only get the right answers but must also do neat, complete and concise writeups that make obvious what each problem is, how you're solving the problem and what your answer is. You also need to include appropriate graphs and tables.

In the last group of Investigation we introduced trees and some of their terminology. The objective of this Investigation is to introduce some applications of trees including binary search and decision trees.

1. We begin with binary search trees. Make use of the fact that the following are **binary search trees**

   and the following are not

   to explain when a tree is a binary search tree

2. From Problem (1) we know that every internal vertex of a binary search tree has at most two children. In particular, at most one *left child* (drawn to the left) and at most one *right child* (drawn to the right)
   a. Draw a graph that is a binary search tree
   b. Draw a graph that is not a binary search tree
   c. What does the binary in binary search tree refer to

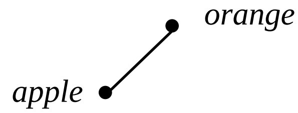
3. The objective of this problem is to illustrate how to make a **binary search tree** - a tree that puts elements from a list like the following

   orange, apple, banana, peach, avocado, apricot, plum, persimmon

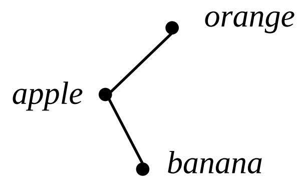
   in an order that makes it easy to search for their presence. We put this list of elements in "alphabetic" order as follows

   (1) *orange*: Since *orange* is the first member of the list we make it the root as follows

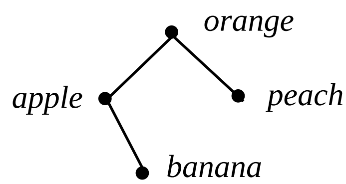
   (2) *apple*: Since *apple* comes before orange we make *apple* its left child as follows



(3) *banana*: Since *banana* comes before *orange* we move along the left to *apple*. Since *banana* comes after *apple* we make banana the right child of *apple* as follows



(4) *peach*: Since *peach* comes after orange we make it the right child of *orange* as follows



Note that we always start at the root and work our way down to find the next element in the list

a. Complete the binary search tree for the list above
b. Explain how the descendants on the left branch of a vertex are different from the descendants on the right side

4. Find the binary search tree for the numbers

$$7, \ 3, \ 9, \ 6, \ 8, \ 0$$

arranged in "numerical" order

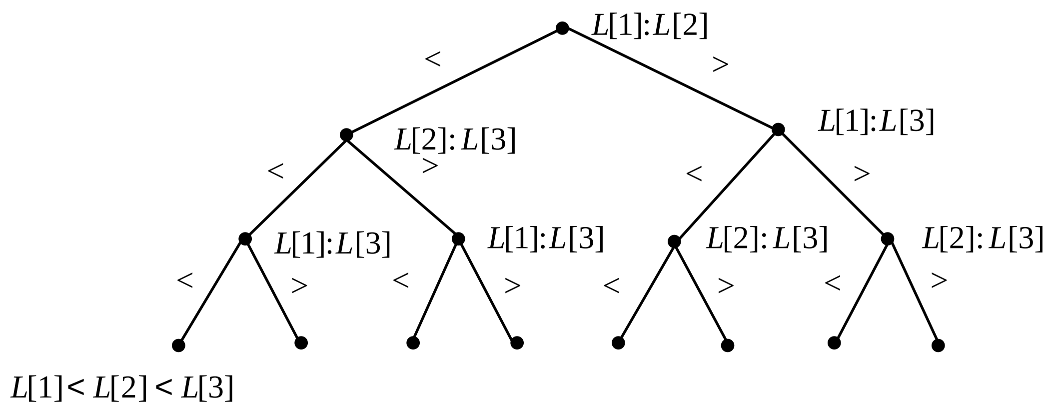
5. Do your own example of a binary search tree

6. As we said in Problem (3) the purpose of creating binary search trees is to make it easier to determine if an element is in a given list
   a. How would you make use of a binary search tree to see if an element is in a given list
   b. Do an example to illustrate finding an element in a given list of at least 10 elements
   c. Do an example to illustrate what happens when an element is not in a given list of at least 10 elements

7. Suppose we have a list L of $n = 2^{10}$ elements
   a. What's the average number of comparisons it would take to verify that an element e is in L by brute force - by starting at the beginning and comparing e to every element in the list
   b. What's the most comparisons it would take to search a binary search tree to verify that e is in the list
   c. How much faster are binary search trees than brute force

8. The objective of this problem is to introduce decision trees. The purpose of **decision trees** is to speed up finding things out like the relative sizes of different numbers. Suppose, in particular, we want to find the relative sizes of the three different numbers $L[1]$, $L[2]$ and $L[3]$.

A decision tree will speed up determining which of the following is true: $L[1] < L[2] < L[3]$, . . . , $L[3] < L[2] < L[1]$. Here is the corresponding decision tree



a. Explain what's going on in this decision tree
b. Complete the bottom line of the decision tree. Be sure to identify which vertices correspond to impossible situations.

9. Draw subsets of the decision tree from Problem (8) corresponding to finding the relative orders of the numbers
   a. 7, 3, 8
   b. 8, 7, 3

10. How many levels are required in a decision tree for comparing four numbers