* Start from beginning.
* Symbol table ADT
  + Motivating example: bank account balances
  + int acct number -> double balance.
  + struct { int, double }.
  + 2 -> $40
  + 3 -> $50
  + 5 -> $10
  + 7 -> $80
  + 9 -> $30
* Show symbol table as unsorted array.
* Do big-oh of contains, insert, delete.
* Show symbol table as sorted array.
* Do big-oh
* Show symbol table as BST
* Do big-oh.
* Remember, BSTs are a specific type of BINTREE, which is a specific type of TREE.
  + draw venn diagram
* Review contains/insert/delete algorithms (hand wavy).
* Practice insert/delete.  
  + Show how to insert 4, 6, 9, 2, 3 and 7 into an initially empty binary search tree. (Show each step.)
  + Delete root.
* Review insert/delete/contains code.
* Big-oh of insert/delete/contains. T(n) = T(n/2)+1, T(1) = 1
* Recursive functions on trees.

Show class size-of-tree function (aka count nodes).  
  
/\*   
 Compute the number of nodes in a tree.   
\*/   
int size(struct node\* node) {   
  if (node==NULL) {   
    return(0);   
  } else {   
    return(size(node->left) + 1 + size(node->right));   
  }   
}

* Show class height function.
* /\*   
   Compute the "maxDepth" of a tree -- the number of nodes along   
   the longest path from the root node down to the farthest leaf node.   
  \*/   
  int maxDepth(struct node\* node) {   
    if (node==NULL) {   
      return(0);   
    }   
    else {   
      // compute the depth of each subtree   
      int lDepth = maxDepth(node->left);   
      int rDepth = maxDepth(node->right);
* // use the larger one   
      if (lDepth > rDepth) return(lDepth+1);   
      else return(rDepth+1);   
    }   
  }

* Class writes sum of nodes.
* height of specific node in BST.
* minValue in BST/regular bintree
* Wrapup.
* Talk about in C++.
* map<int, double> accounts.
* accounts[acct#] = balance. INSERT

mymap.erase ('c'); REMOVE

**CONTAINS**: std::map<char,int> mymap;

std::map<char,int>::iterator it;

mymap['a']=50;

mymap['b']=100;

mymap['c']=150;

mymap['d']=200;

it = mymap.find('b');

if (it != mymap.end())

mymap.erase (it);