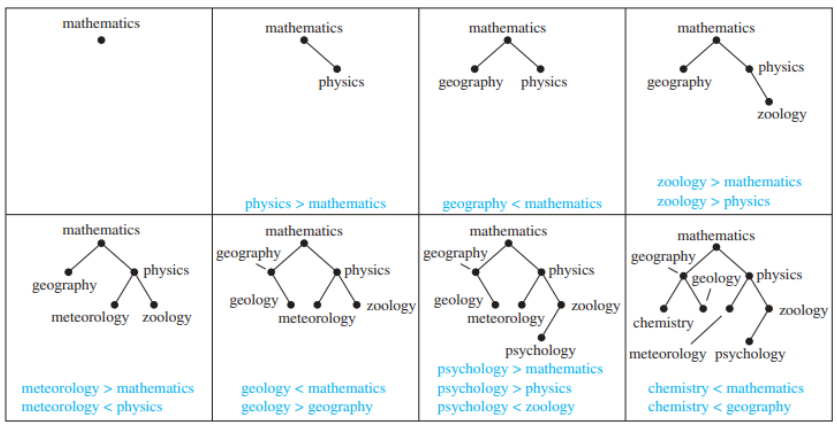
Ch11.2 Applications of Trees (Week 16)

How should items in a list be stored to that an item can be easily located?

What series of decision should be made to find an object with a certain property in a collection of objects of a certain type?

Binary Search Trees

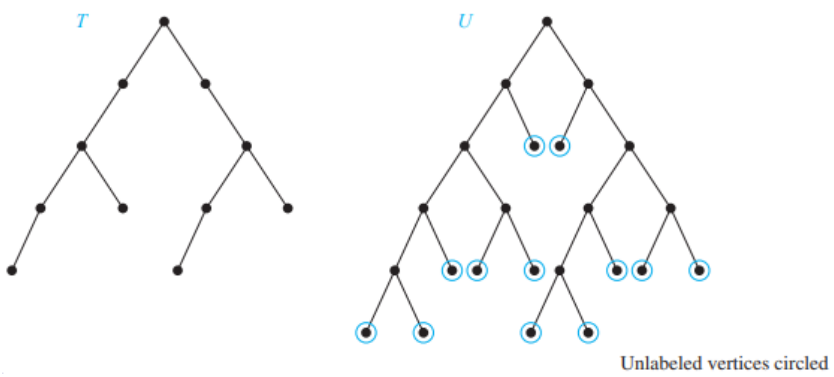
→ each vertex is labeled with a key, compare the keys already in the tree, > go right, < go left



Form a binary search tree using alphabetical order

ALGORITHM 1 Locating an Item in or Adding an Item to a Binary Search Tree.

procedure *insertion*(T: binary search tree, x: item)
v := root of T
{a vertex not present in T has the value *null*}
while $v \neq null$ and $label(v) \neq x$
  if $x < label(v)$ **then**
    if left child of $v \neq null$ **then** v := left child of v
    else add *new vertex* as a left child of v and set $v := null$
  else
    if right child of $v \neq null$ **then** v := right child of v
    else add *new vertex* as a right child of v and set $v := null$
if root of $T = null$ **then** add a vertex v to the tree and label it with x
else if v is null or $label(v) \neq x$ **then** label *new vertex* with x and let v be this new vertex
return v {v = location of x}



Adding unlabeled vertices to make a binary search tree full