Binary search trees

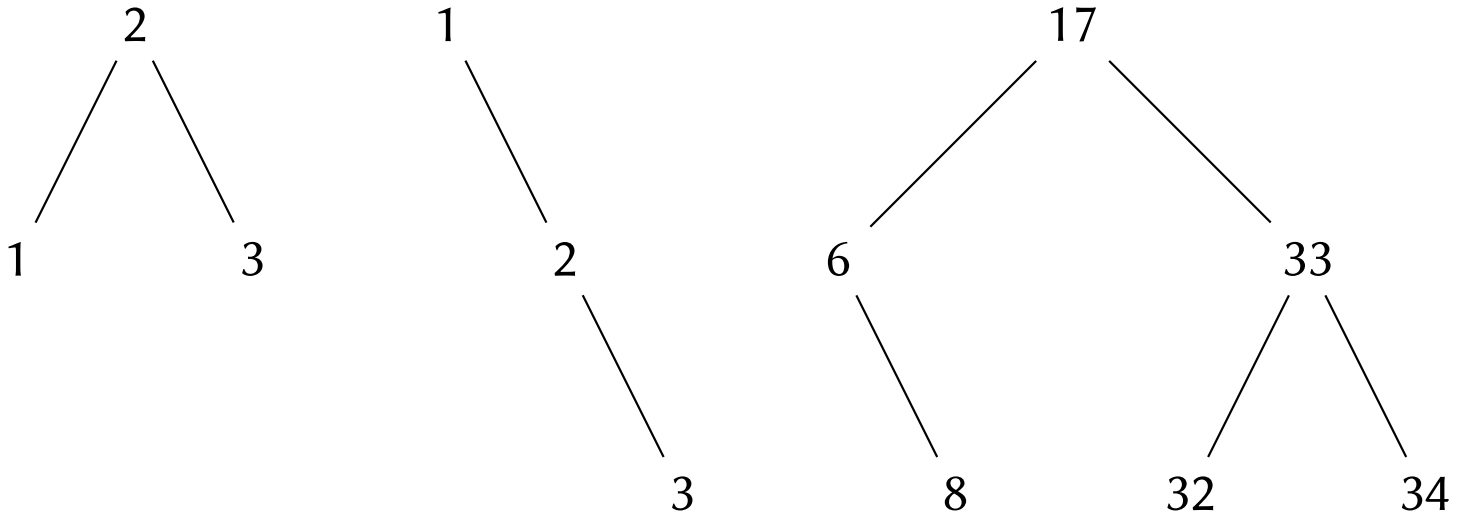
Goldsmiths Computing

December 10, 2018

Binary search tree property

Let x be a node in a binary search tree. If y is a node in the left subtree of x, then y.key $<$ x.key. If z is a node in the right subtree of x, then z.key $\geq$ x.key.

Example binary search trees



Find in binary tree

```
Require: tree :: binary tree
  function FIND(tree,object)
      if NULL?(tree) then
          return false
      end if
      if tree.key = object then
          return true
      end if
      return FIND(tree.left,object) ∨ FIND(tree.right,object)
  end function
```

Complexity analysis

find

$$T(N) = T(k) + T(N - k - 1) + \Theta(1)$$

$$\Longrightarrow \Theta(N)$$

max

(as with find: traverse all nodes)

Find in binary search tree

```
Require: tree :: binary search tree
  function FIND(tree,object)
      if NULL?(tree) then
          return false
      end if
      if tree.key = object then
          return true
      else if tree.key > object then
          return FIND(tree.left,object)
      else
          return FIND(tree.right,object)
      end if
  end function
```

Complexity analysis

Work in terms of the height h of the tree

find

key to find could in principle be on the lowest level of the tree

$$\Longrightarrow \Theta(h)$$

max

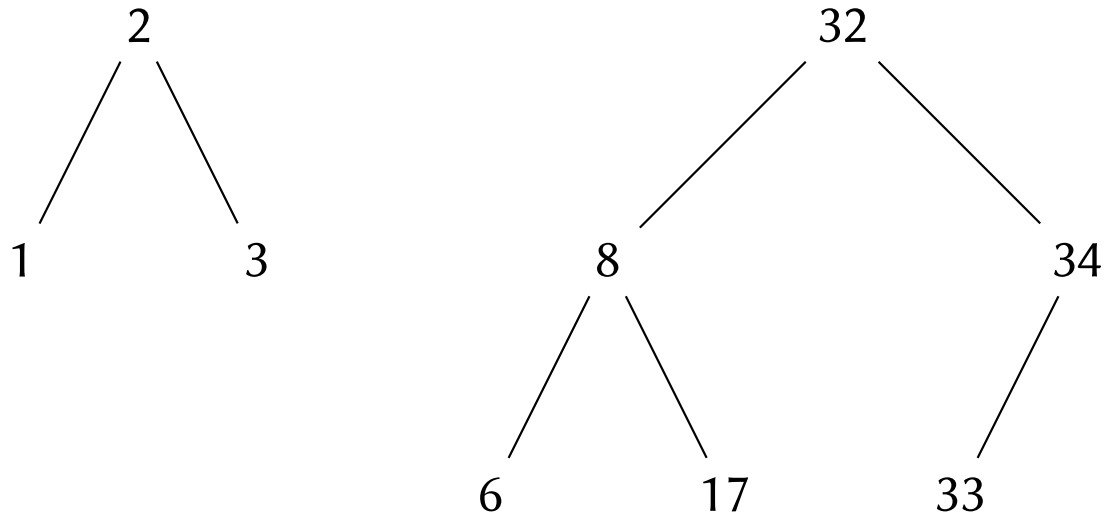
Descend right subtrees as far as possible

$$\Longrightarrow \Theta(h)$$

Nearly complete binary search trees

A nearly complete binary search tree has both the binary search tree property and the complete property.

Example nearly complete binary search trees



Complexity analysis

For a complete binary search tree, h is $\lceil \log(N) \rceil$.

max

traverse just right-nodes

$$\Longrightarrow \Theta(\log(N))$$

find

$$T(N) = T\left(\frac{N}{2}\right) + \Theta(1)$$

$$\Longrightarrow ...?$$

constructor

$$\Longrightarrow \Theta(N \log(N))$$

Work

1. Reading
   - CLRS, section 10.4
   - CLRS, chapter 12
2. Questions from CLRS:

   Exercise 10.4-1

   Exercises 12.1-1, 12.2-1, 12.2-2, 12.3-1, 12.3-3