

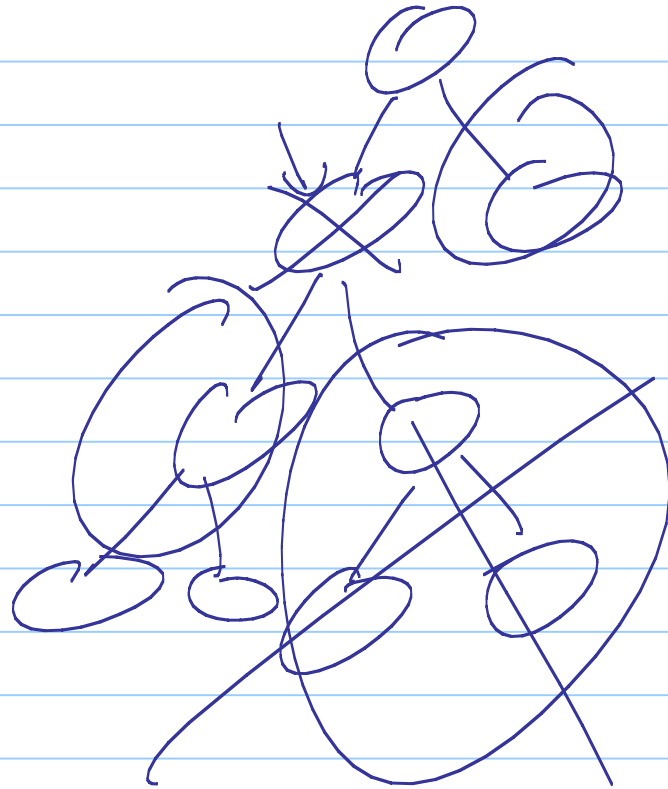
CS 180 - Search Trees

Note Title

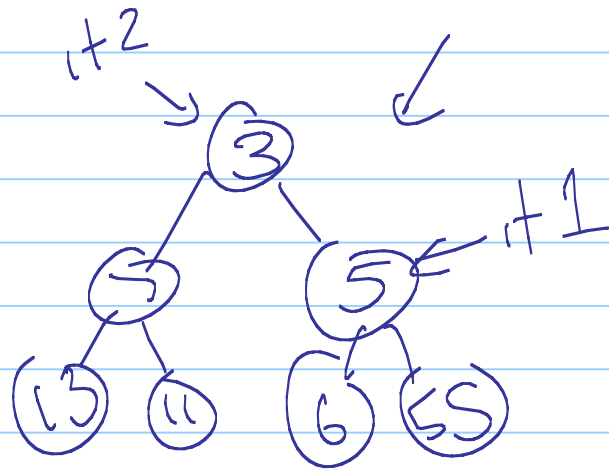
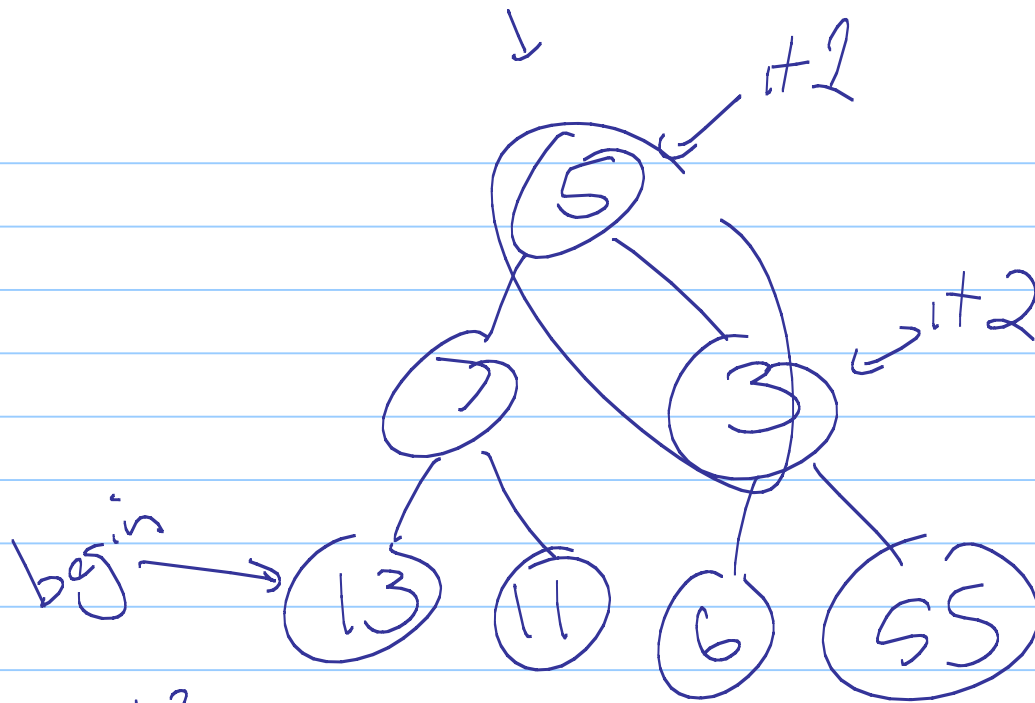
11/9/2009

Announcements

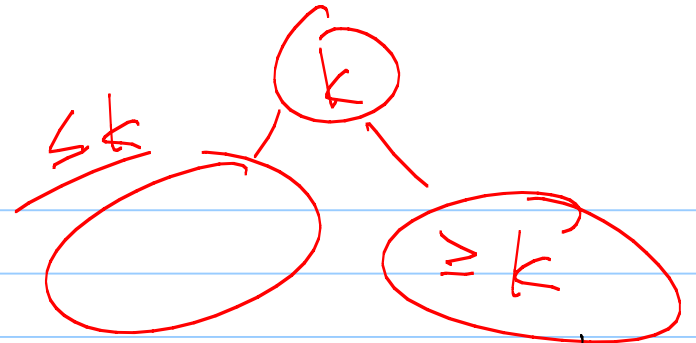
- ~~HW~~ due tomorrow
Program
- HW is coming - due next Friday
- Midterm 2 in 1 week from next Monday
(or later (Nov. 8) that week)



Pivot



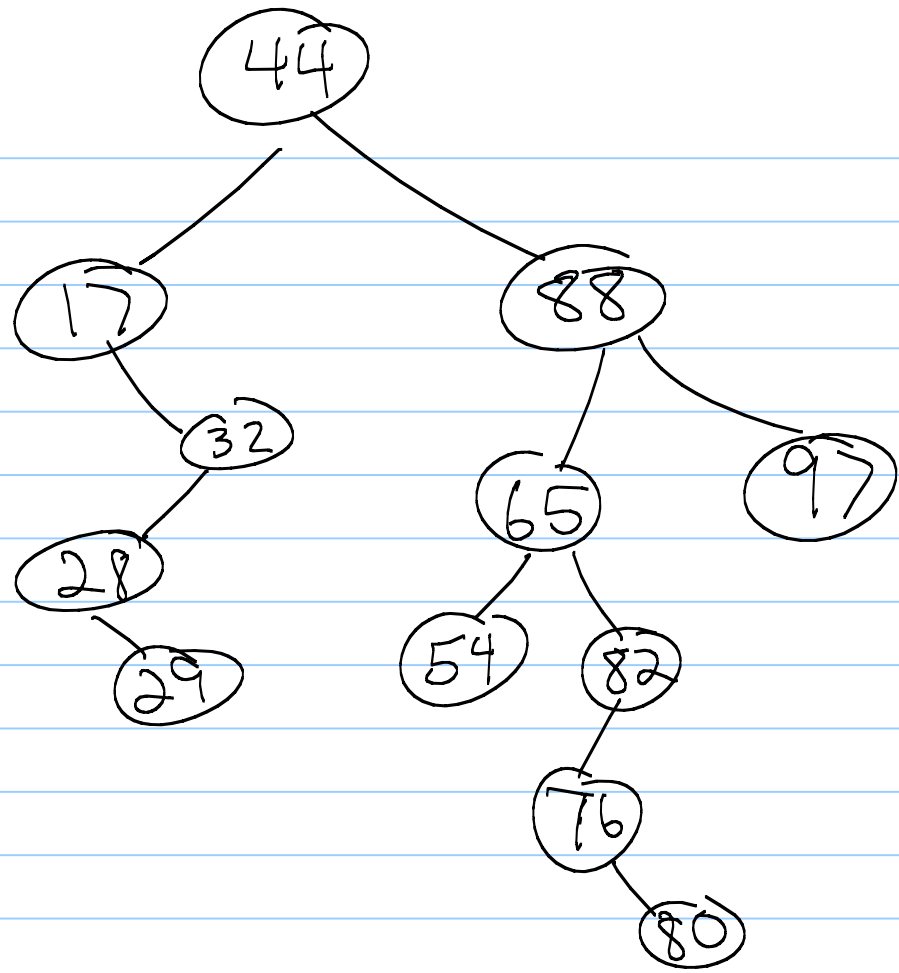
Binary Search Trees



A binary tree such that each internal node v of T stores a key k , and:

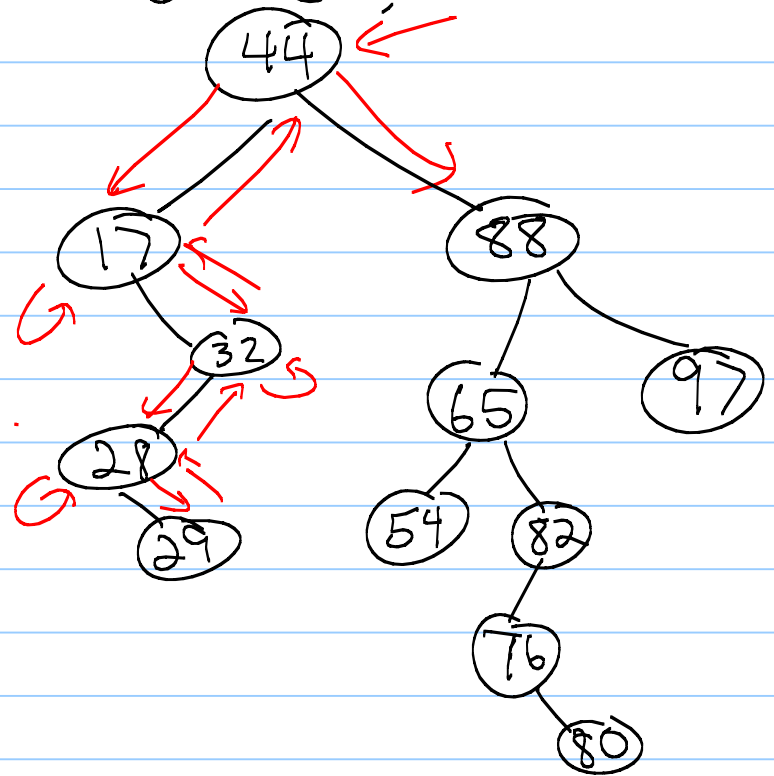
- keys stored at nodes in the left subtree of v are less than or equal to v
- keys stored in the right subtree are greater than or equal to v .

Is 68 in
the tree?



~~pre order~~
~~post order~~
→ in order

Q: What type of traversal will print the elements in sorted order?

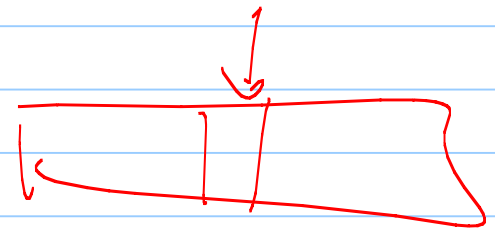


17-28-29-32-44

How do we search for an element ^x in the tree?

```
check root.  
if root < x  
    search right  
if root > x  
    search left  
else return yeah!
```

+ base case



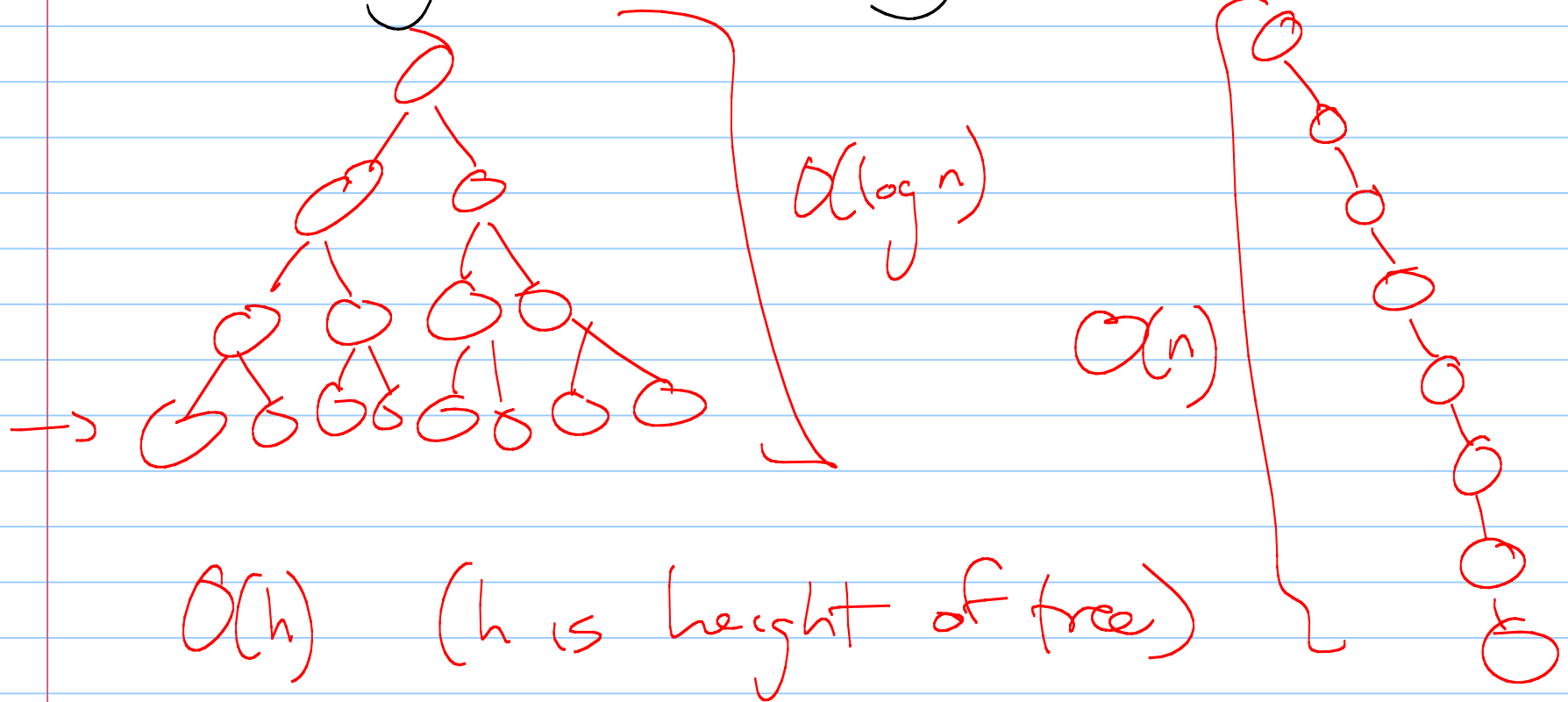
TreeSearch(k, v):

Input: key k to search for & a node v in tree

Output: A node w of T s.t. either w holds key k or w is the leaf node where k would belong

```
( if key(v) == k
  return v
  else if key(v) < k
    → TreeSearch(k, v → left)
    else if key(v) > k
      → TreeSearch(k, v → right)
    else // at a leaf
      → return v
```

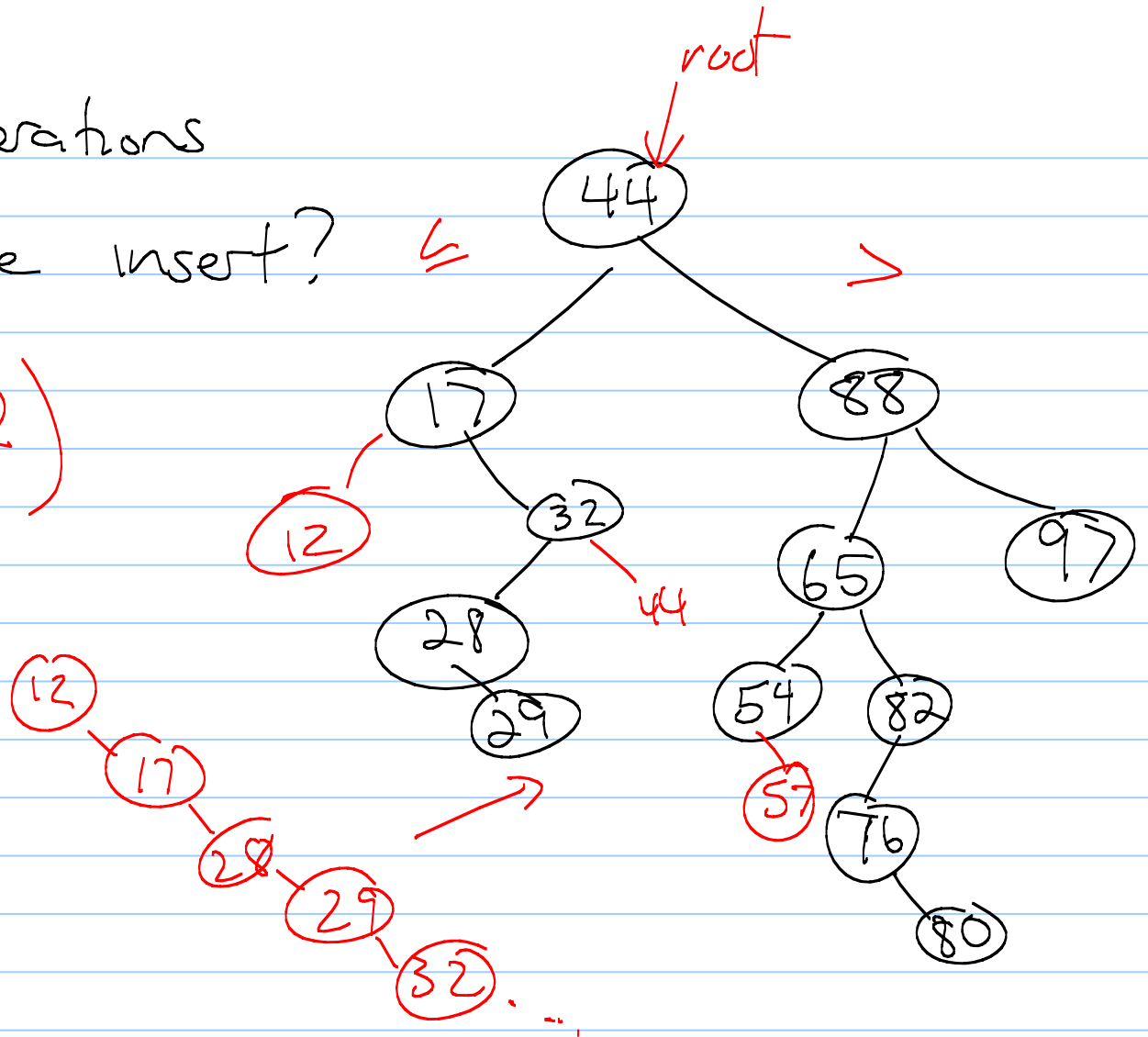

How long does searching take?



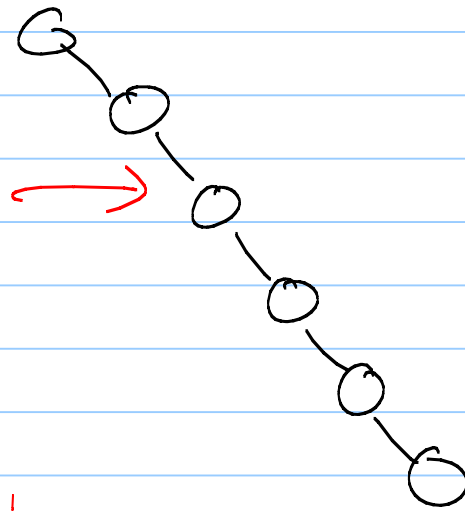
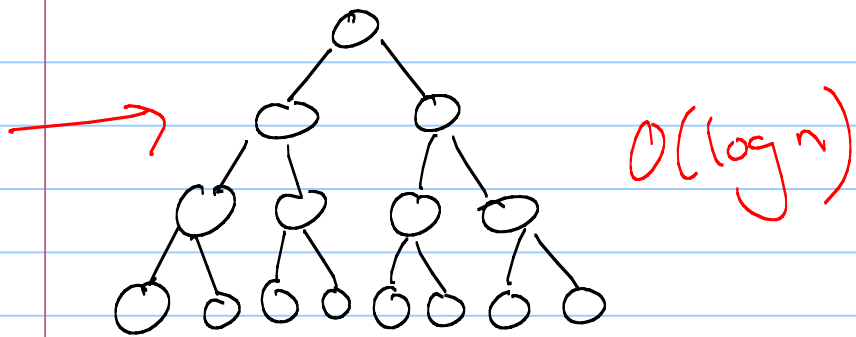
Update Operations

- How do we insert?

Insert (12)
insert (57)



Best versus worst case:



$$2^0 + 2^1 + \dots + 2^h = n$$
$$\Rightarrow 2^{h+1} - 1 = n \Rightarrow h \approx \log n$$

How deep is each tree (in terms of $n = \#$ of nodes in tree)?