

# MA/CSSE 473

## Day 24

**2-3 Trees**

**Heaps and HeapSort**

**Problem Reduction**



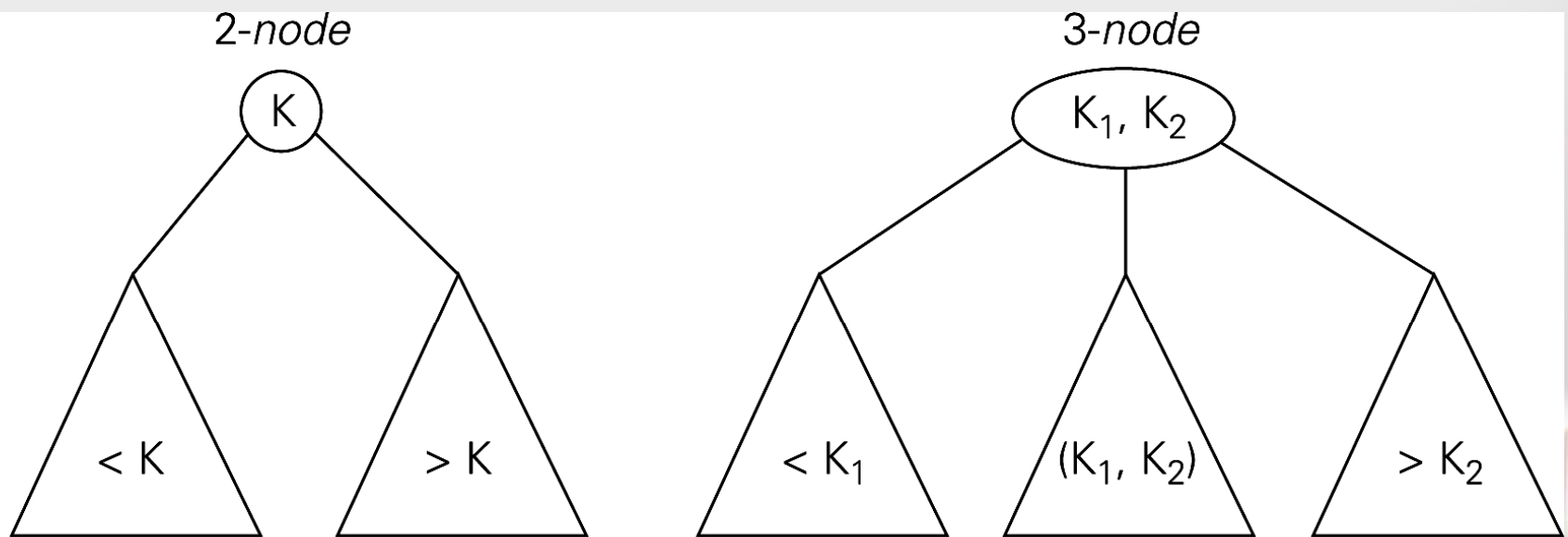
# MA/CSSE 473 Day 24

- HW 10 due tomorrow at noon
  - Bring hard copy to my office, or leave it in my mailbox in F-233
- HW 11 will be due Friday, Oct 24
- Exam 2, Friday, Oct 31
- **Student Questions**
- Transform and conquer
  - Most examples should be review.



# 2-3 trees

- Another approach to balanced trees
- Keeps all leaves on the same level
- Some non-leaf nodes have 2 keys and 3 subtrees
- Others are regular binary nodes.

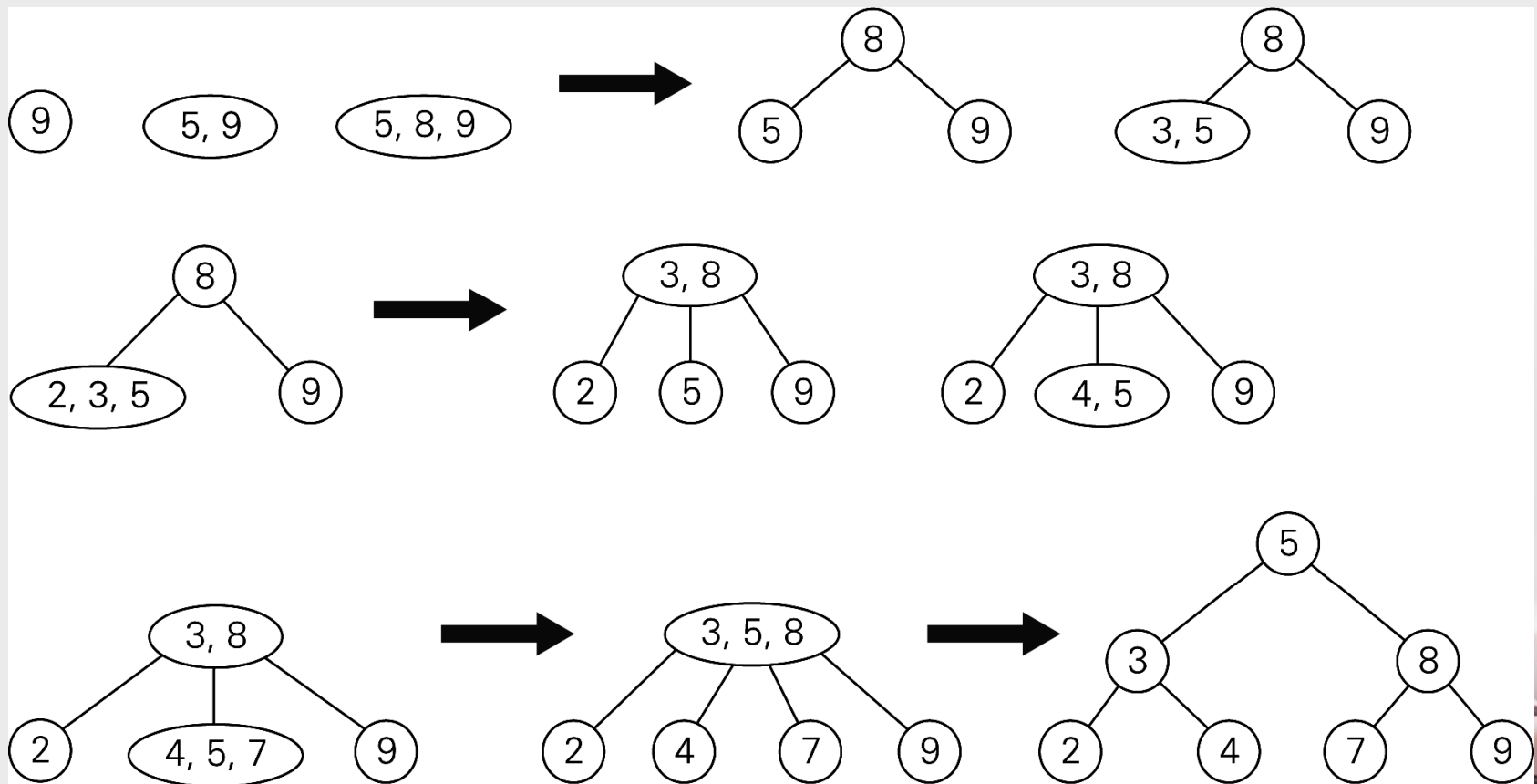


**FIGURE 6.7** Two kinds of nodes of a 2-3 tree

# 2-3 tree insertion example

- More examples of insertion and deletion:

[http://www.cs.ucr.edu/cs14/cs14\\_06win/slides/2-3\\_trees\\_covered.pdf](http://www.cs.ucr.edu/cs14/cs14_06win/slides/2-3_trees_covered.pdf)



**FIGURE 6.8** Construction of a 2-3 tree for the list 9, 5, 8, 3, 2, 4, 7

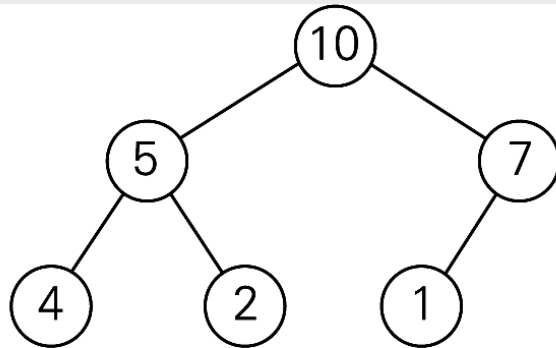
# Efficiency of 2-3 tree insertion

- Upper and lower bounds on height of a tree with  $n$  elements?
- Insertion time is proportional to the height of the tree.



# Binary (max) Heap Quick Review

- An almost-complete Binary Tree
  - All levels, except possibly the last, are full
  - On the last level all nodes are as far left as possible
  - No parent is smaller than either of its children
  - A great way to represent a Priority Queue
- Representing a binary heap as an array:



the array representation

index	0	1	2	3	4	5	6
value		10	5	7	4	2	1
		parents			leaves		

**FIGURE 6.10** Heap and its array representation

# Insertion and RemoveMax

- Insertion:
  - Insert at the next position to maintain an almost-complete tree, then percolate up to restore heap property.
- RemoveMax:
  - Move last element of the heap to the root, then percolate down to restore heap property.
- Both operations are  $\Theta(\log n)$ .
- Demo:  
<http://www.cs.auckland.ac.nz/software/AlgAnim/heaps.html>



# HeapSort

- Arrange array into a heap
- Starting from the root, remove each element from the heap and move to the end of the array.
- Animation:  
<http://www.cs.auckland.ac.nz/software/AlgAnim/heapsort.html>
- Faster heap building algorithm: **buildheap**  
[http://students.ceid.upatras.gr/~perisian/data structure/HeapSort/heap\\_applet.html](http://students.ceid.upatras.gr/~perisian/data_structure/HeapSort/heap_applet.html)

