

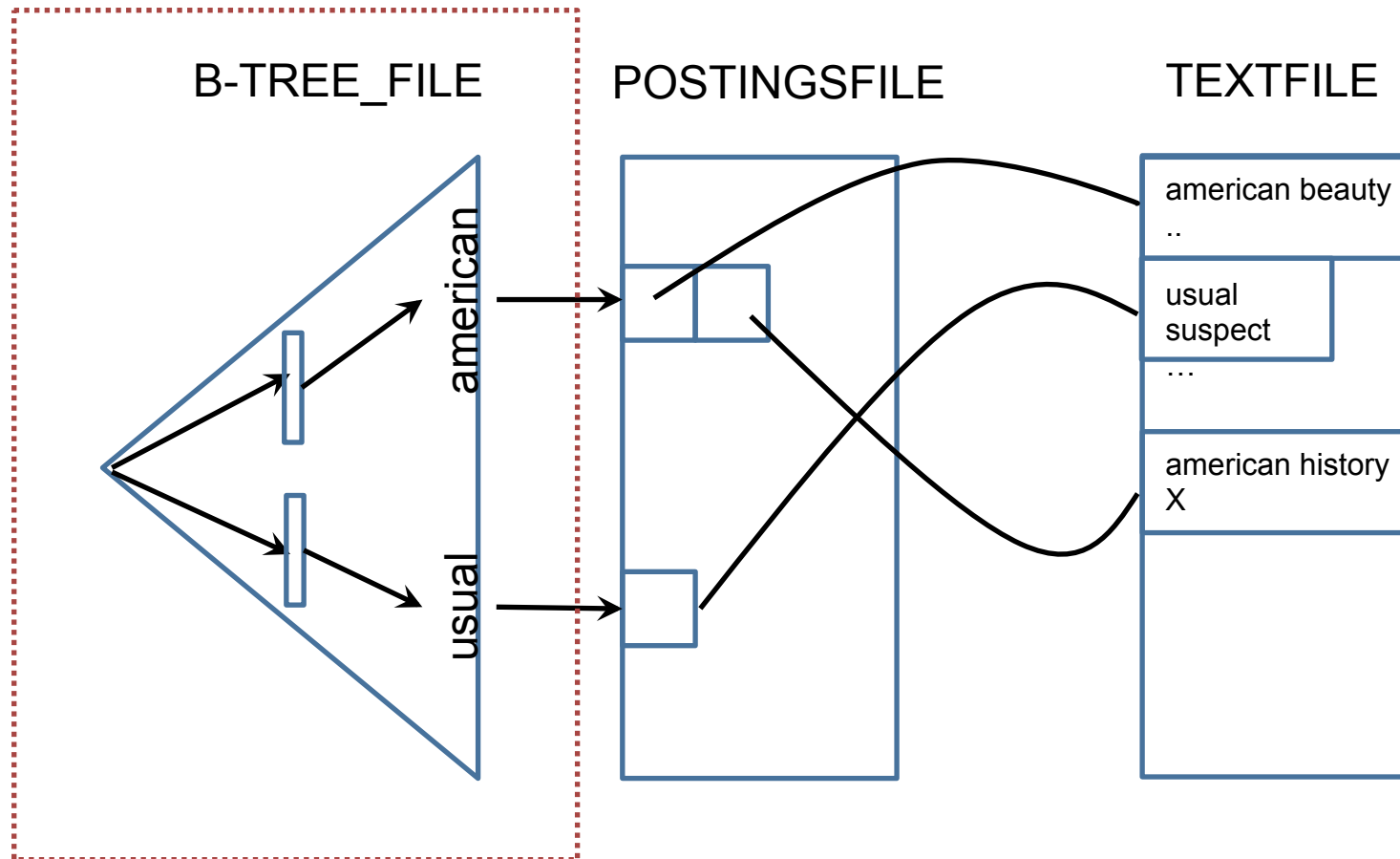
**15-415/615**  
**Database Applications**  
**Fall 2016**

**HW3: B+ Tree Structure**

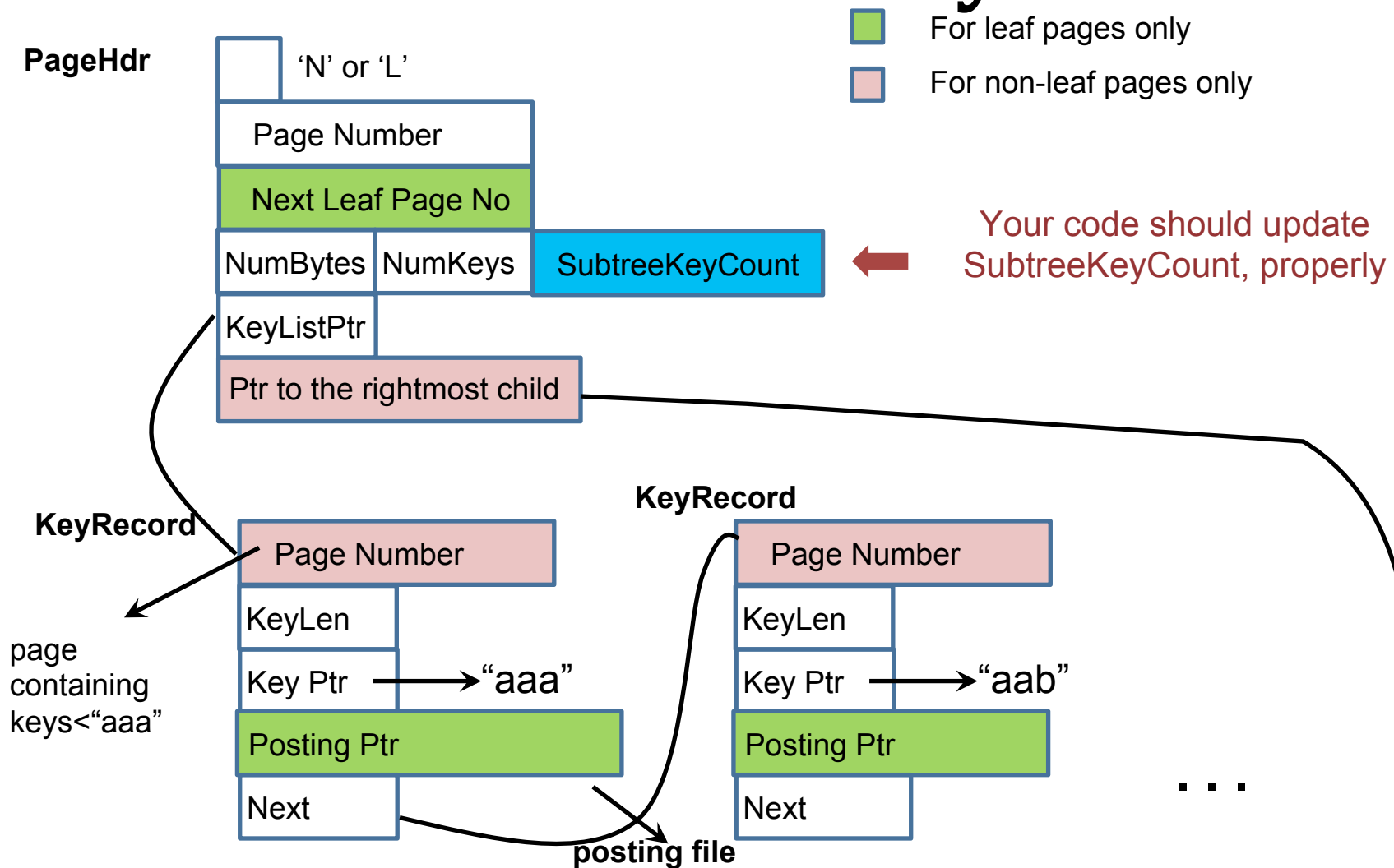
Huanchen Zhang  
Carnegie Mellon University

# B+ Tree Structure, on disk

Your modifications: will be here:



# Structure of a Page (def.h), in memory

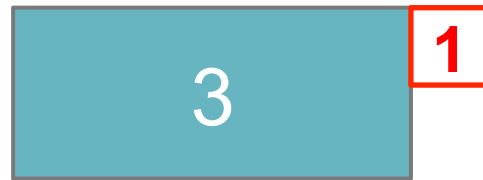


# Insertion Visualized

- insert [3, 7, 13, 9, 1, 6] into an empty tree
- Assume that the splits are done properly
- Notice the 'red' counter (that **your** code has to update properly)

# Insertion: 3 -> []

Root



**SubtreeKeyCount**

Root:

Internal

Leaf:

# Insertion: 7 -> [3]

Root

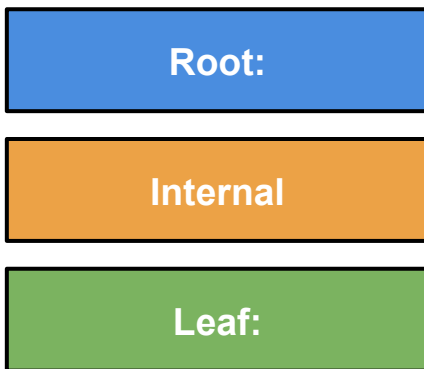


Root:

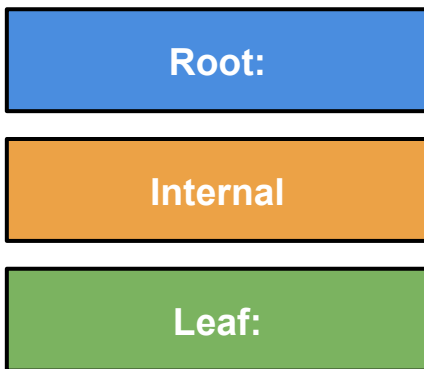
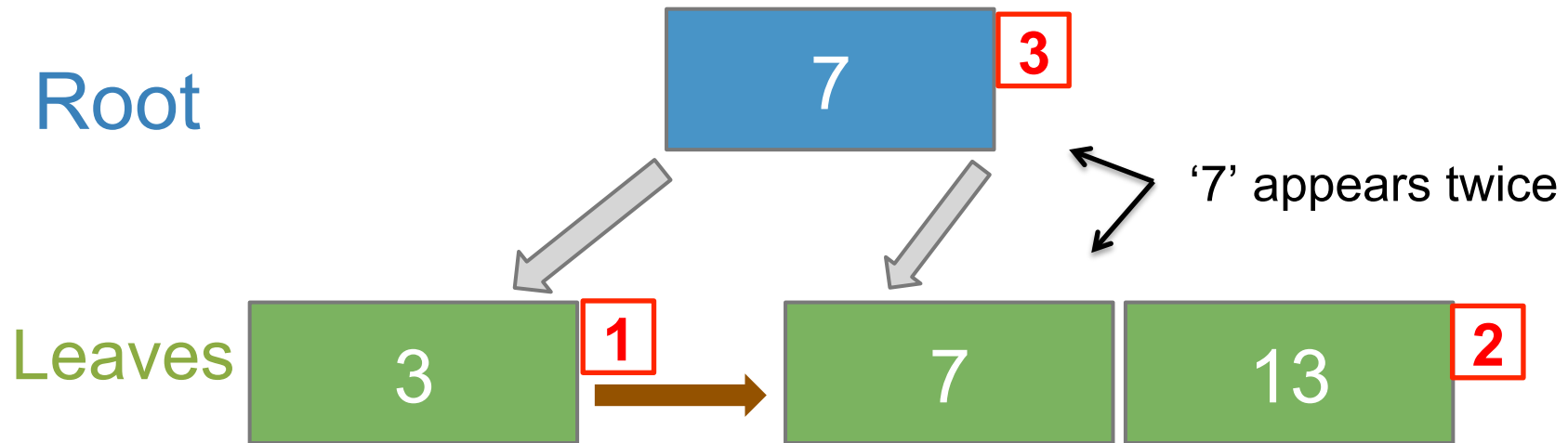
Internal

Leaf:

# Insertion: 13 -> [3, 7]

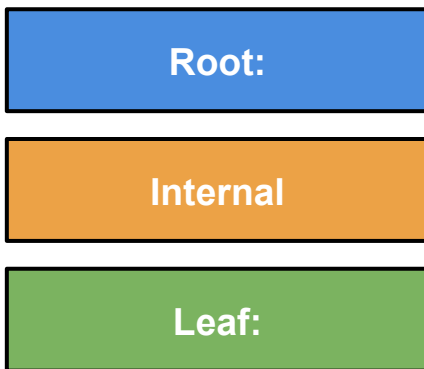
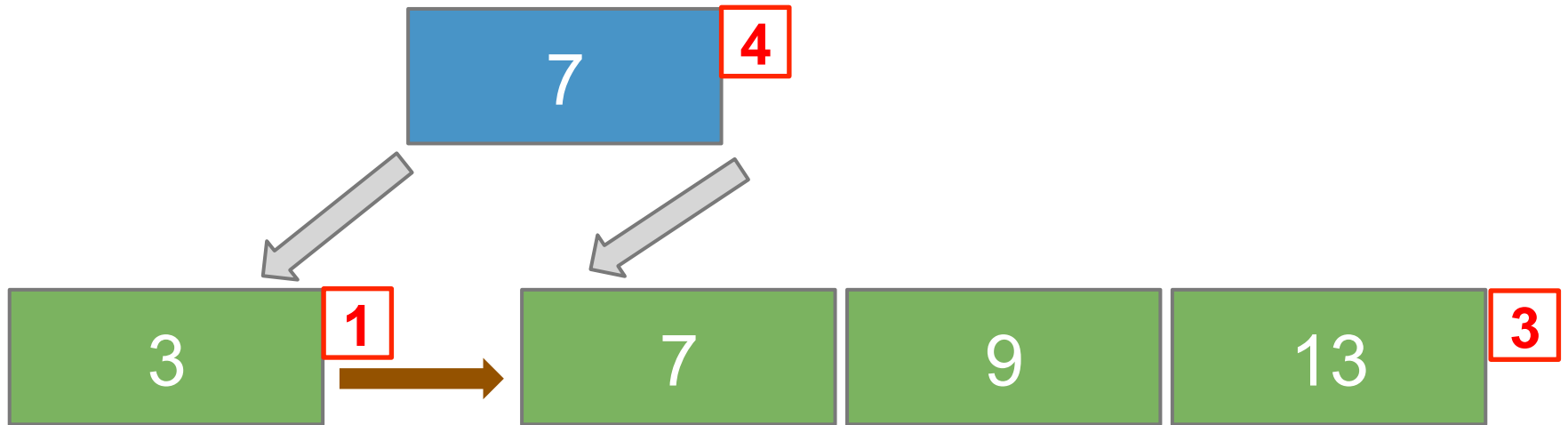


# Insertion: 13 -> [3, 7]

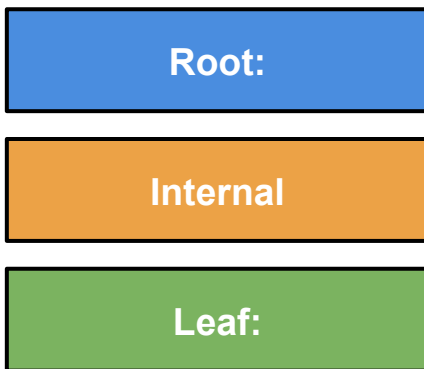
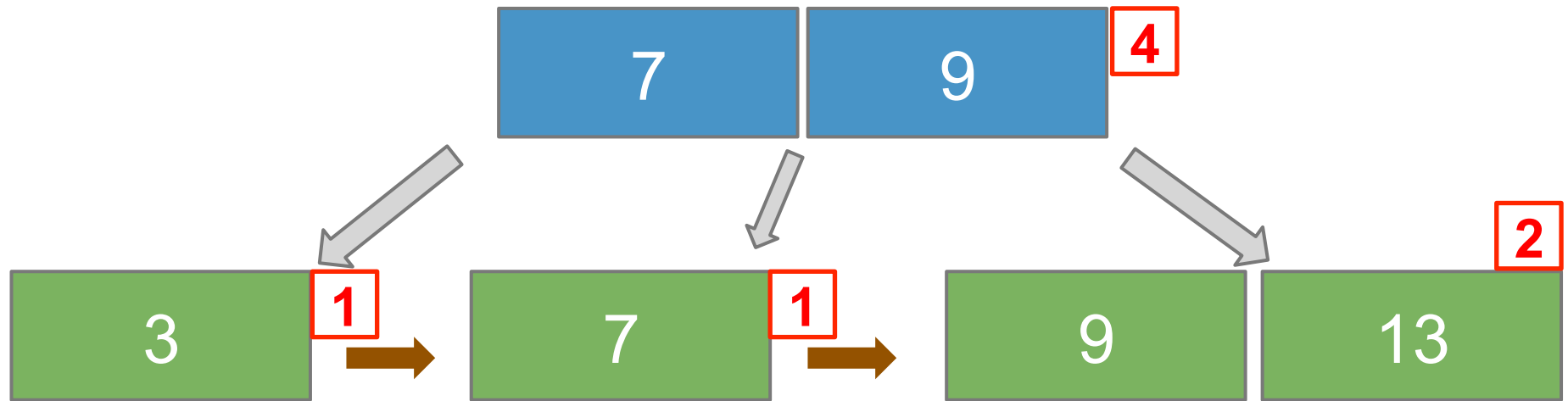




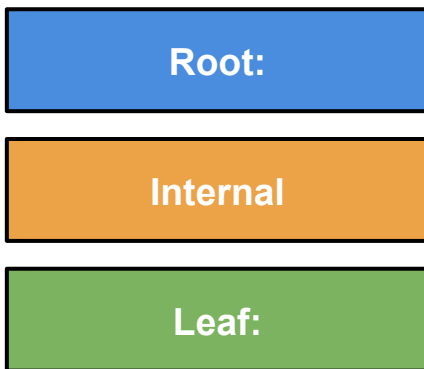
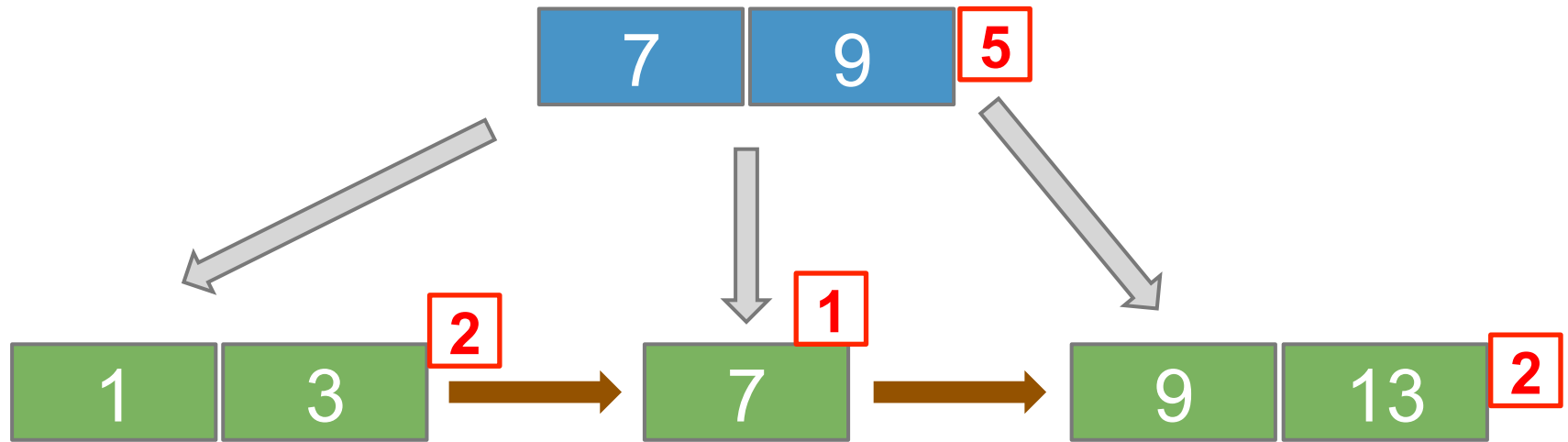
# Insertion: 9 -> [3, 7, 13]



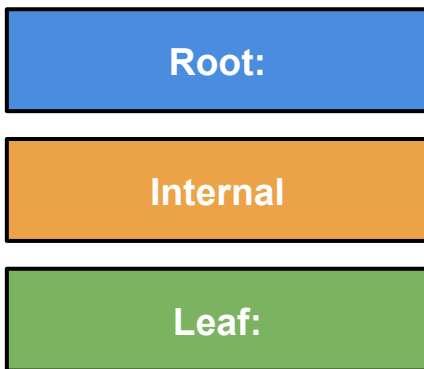
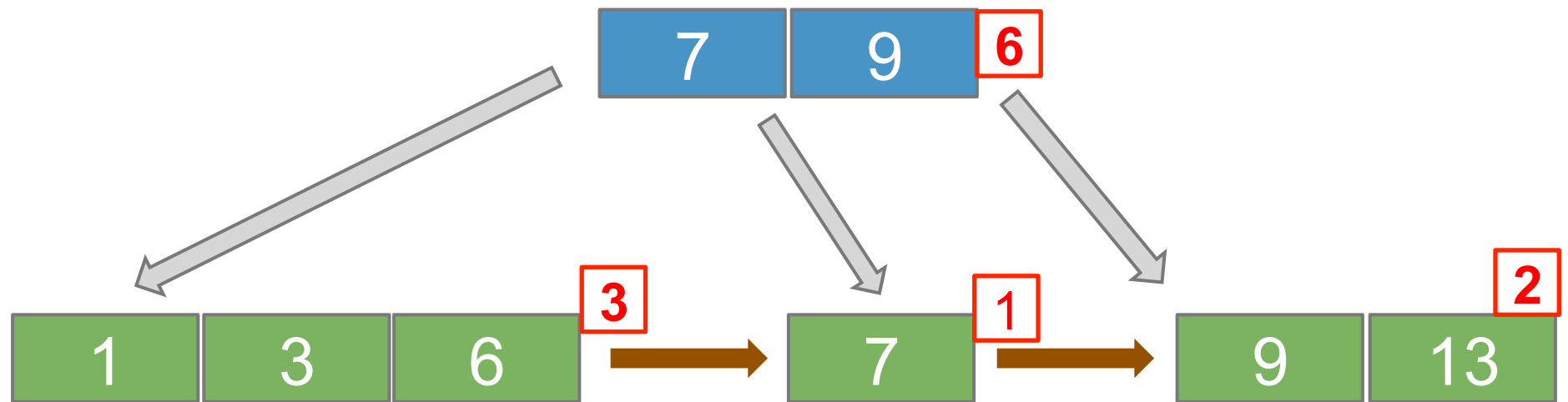
# Insertion: 9 -> [3, 7, 13]



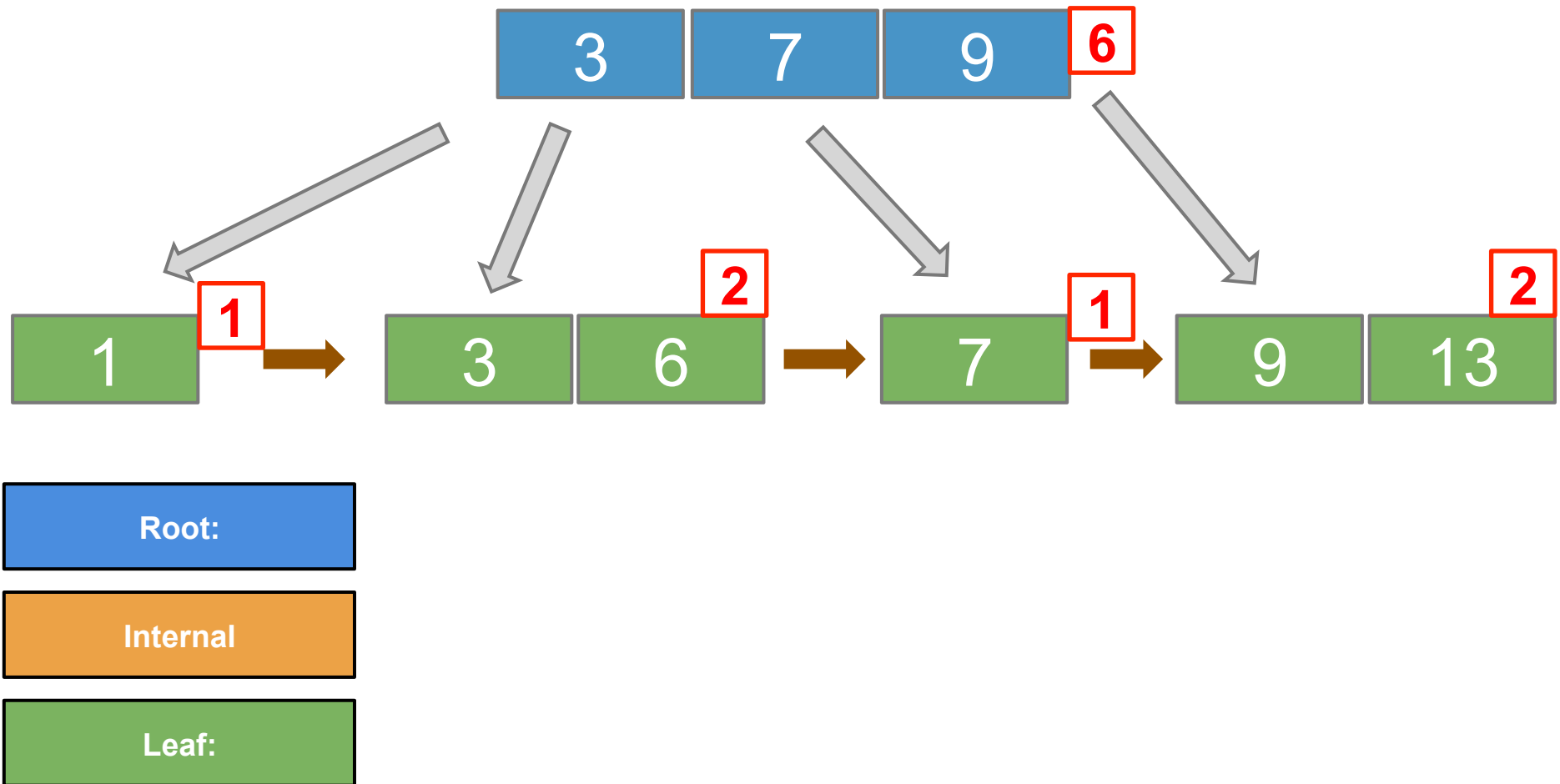
# Insertion: 1 -> [3, 7, 9, 13]



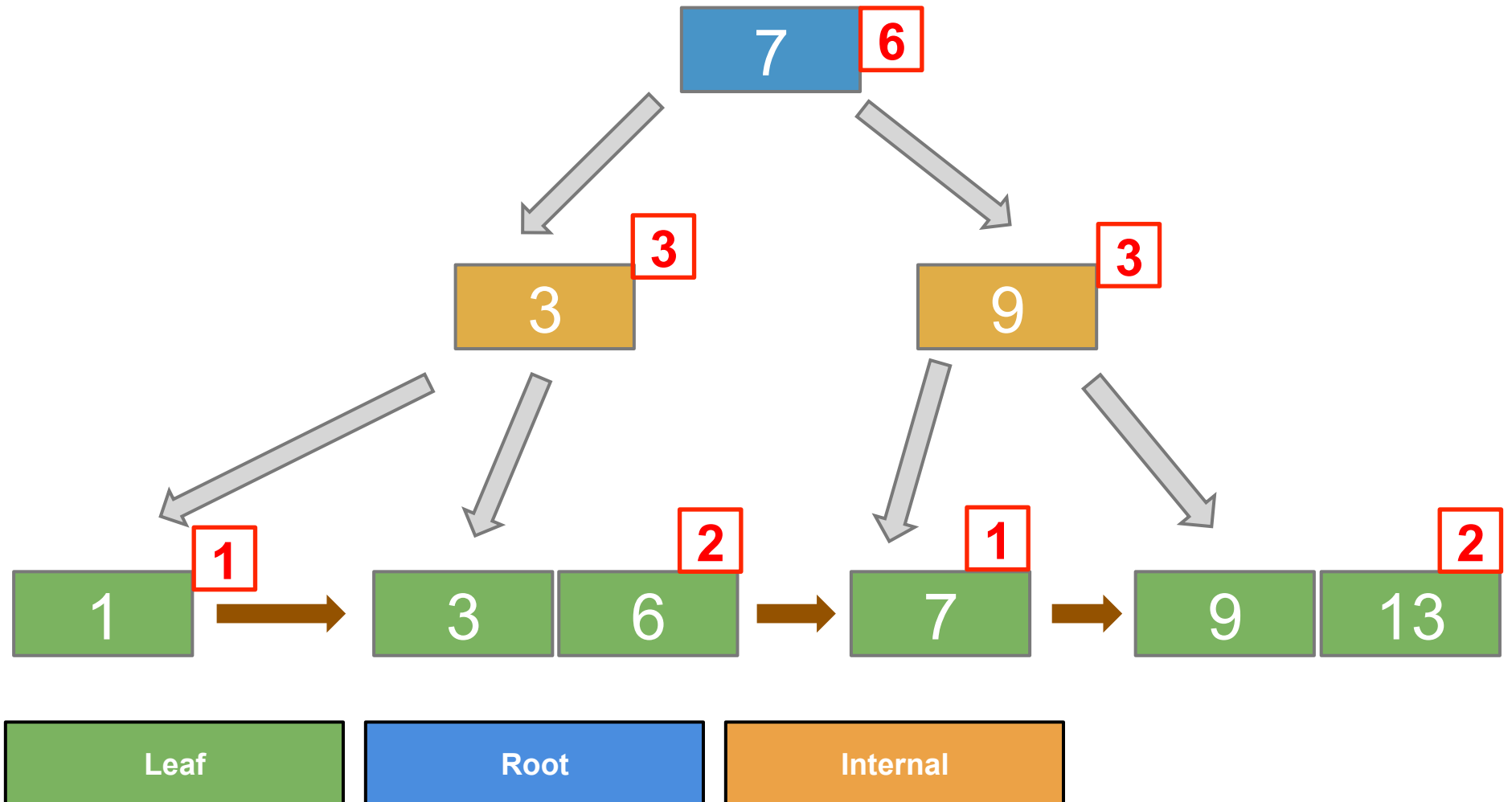
# Insertion: 6 -> [1, 3, 7, 9, 13]



# Insertion: 6 -> [1, 3, 7, 9, 13]



# Insertion: [1, 3, 6, 7, 9, 13]



# Questions?

- Come to office hours (5 TAs + instructors)
- Understand the provided code before starting
- Post your questions on blackboard
- Check the webpage for clarifications
- **Start early**
- **PLEASE, start early**