# 15-110 recitation 08

## Recap

- trees, binary search trees
- dictionaries, hashing
- sorting

## Reminders!

- HW 4 check-in due Monday!

# Problems

## mergesort review

**Problem**

Trace the sort on this list [7, 5, 2, 3, 9, 1]
What is the result of the first divide? Second? First merge?
How many steps does mergesort take? What is its complexity?
Does the runtime improve if the list is already sorted in ascending order? Descending order?

Trace the sort:

How many steps does each *pass* of merge sort take?

How many *passes* does mergesort take?

What is the overall complexity of mergesort?

## mostWins

| | |
|---|---|
| **Problem** | Given a list of wins by CMU, Pitt, OSU, PennState's, and another unspecified number of football teams, return the team with the most wins. |

```
def mostWins (L):
```

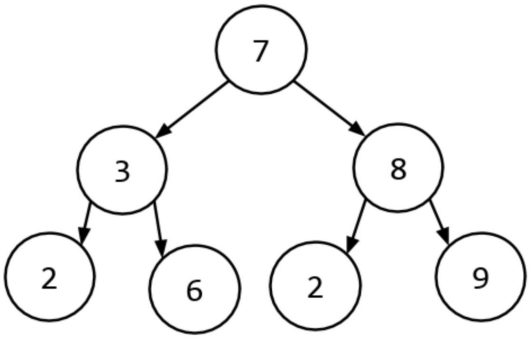
## trees

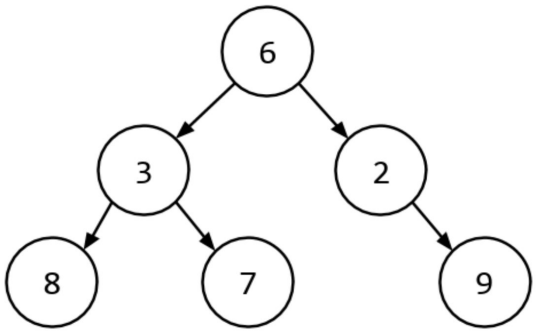
| | |
|---|---|
| **Problem** | **[5 min] printLeaves**- write a function given a dictionary representation of a tree, prints the leaves of the tree.<br>**[10 min]** countNodes- write a function given a dictionary representation of a binary tree, prints the number of nodes in the tree<br>**[5 min]** motivate why search on a BST is O(logn) by alluding to the structure of the tree (this is review from lecture) |

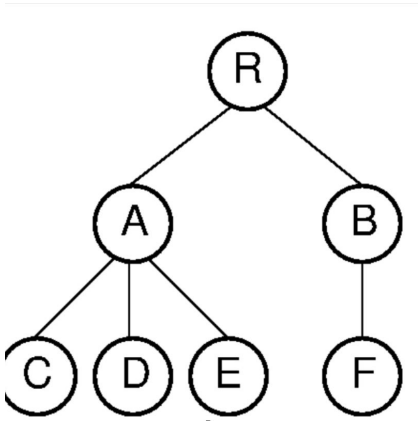
```
def printLeaves(tree):
```

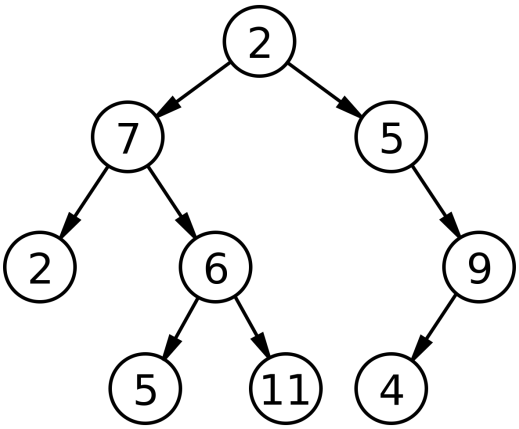
```
def countLeaves(tree):
```

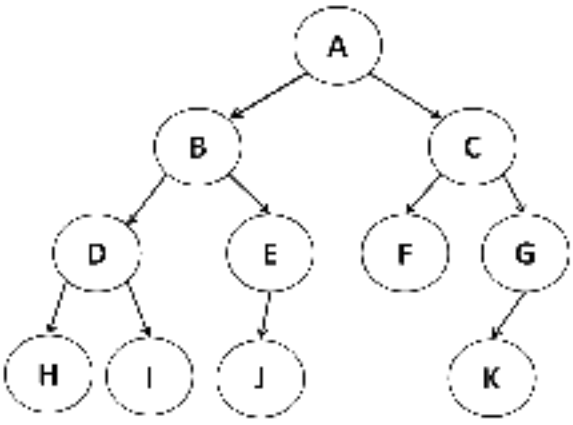
**Not BST**



**BST**