CS16 Section 3
Monday February 24 - Wednesday February 26
Agenda

1. Icebreakers
2. Mini-Assignment
3. Hashing
4. Binary Search Trees
5. Coding Conventions
Mini Assignment - Tree Traversals

Please form groups!

Each team should choose between doing
- Inorder
- Postorder
- Preorder
- Breadth-first Search
1. **Inorder:** M-P-A-E-L-P-Y-S-U-R (left, self, right)
2. **Preorder:** P-E-P-M-A-L-U-Y-S-R (Self, left, right)
3. **Postorder:** M-A-P-L-E-S-Y-R-U-P (left, right, self)
4. **Breadth-First:** P-E-U-P-L-Y-R-M-A-S (each child at each level)
Practice Problem
Pseudocode your solution!

Given two lists $A$ and $B$, how can you determine whether any element of $A$ is an element of $B$?
First Approach

O(A*B) time answer: iterate through both lists in a nested for loop and check if any two elements match

```python
for i in range 0 to length of list A
    for j in range 0 to length of list B
        if A[i] == B[j]
            return true
    return false
```
Second Approach

O(A+B) time answer: Add list A to some sort of hash set, and check each element of B to find if it’s in the set.

```python
for element in A:
    hashSet.add(element)
for element in B:
    if hashSet.contains(element):
        return true
return false
```
Hashing Overview
Hashing - Concept

How do Hashing Functions work?

- Expected one element per bucket but it doesn’t always work
- We would want to use them because they have constant insert/delete/lookup
Differences between Hash set and Hash Table

- **Hash set**: has no keys, just values, and there’s no ordering
- **Hash table**: maps keys to values, there’s no ordering
Questions?
How to tell if a tree is a BST?

1- Perform an inorder traversal and check if the returned numbers are in non-decreasing order
2- Recursive solution
   ● Check if each subtree meets the definition of a BST:
     ○ The left subtree of a node contains only nodes with keys less than the node’s key.
     ○ The right subtree of a node contains only nodes with keys greater than the node’s key.
     ○ Both the left and right subtrees must also be binary search trees.
Pseudocode

```plaintext
function isValid(node, max, min)
    if node is null
        return true
    if node.value > max or node.value < min
        return false
    return isValid(node.left, node.value, min) and isValid(node.right, max, node.value)
```

Original function call: isValid(root, ∞, -∞)
What’s the difference between a Binary Tree and a Binary Search Tree?
Difference between BT and BST

**Binary Tree:** each node has at most two children

**Binary Search Tree:** Binary Tree where left child $<$ parent and right child $>$ parent.
Insertion and Deletion

Add 16?
Insertion and Deletion

Delete 7?
Add 1?
Insertion and Deletion

Delete 12?
Final Solutions
Coding Conventions

● What are some ways in which the ImportanceMethod could be written well?
  ○ How can we factor this method so that there aren’t a bunch of edge cases?

● ImportanceMethodB.java activity