Section 3 Overview

Agenda

- Collaboration Policy
- Tree Traversals
- getImportance Code Review
- Hashing
  - Hash Tables, Hash Sets, Hash Map

Tree Traversals

Answers:
- Inorder: M-P-A-E-L-P-Y-S-U-R (left, self, right)
- Preorder: P-E-P-M-A-L-U-Y-S-R (Self, left, right)
- Postorder: M-A-P-L-E-S-Y-R-U-P (left, right, self)
- Breadth-First: P-E-U-P-L-Y-R-M-A-S (each child at each level)
Pseudocode
Using Hash Sets
Given two lists $A$ and $B$, how can you determine whether any element of $A$ is an element of $B$?

I. $O(a \times b)$ time answer
   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

II. $O(a + b)$ time answer
   for element in $A$:
      hashSet.add(element)
   for element in $B$:
      if hashSet.contains(element):
         return true
   return false

Optional Problems
Identifying Binary Search Tree (BST)
Write an algorithm determine if a tree is a BST

1. Perform an inorder traversal - check if the values are actually in order
2. Recursive solution - check if each subtree is a valid binary search tree

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)