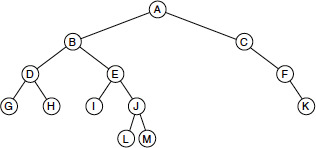
Data Structures and Algorithms  
Week 3 problem sheet

## A. Trees



1. (source: Weiss 18.1) For the tree shown above, state:
   1. Which node is the root?
   2. Which nodes are leaves?
   3. What is the tree’s height?
   4. What are the results of traversing the tree using
      1. preorder
      2. postorder
      3. inorder
      4. level order

## B. Binary trees

1. State whether each of the following statements is TRUE or FALSE, and explain your reasoning.
   1. The height of a binary tree is
   2. The proper descendants of a node’s ancestors are also descendants of that node.
   3. A binary tree always has more external nodes than internal nodes.
   4. Every tree has a root node.



1. (Source: Weiss 18.9) Write efficient methods (and give their Big “O” running times) that take a reference to a binary tree root t and compute
   1. The number of leaves in t
   2. The number of nodes in t that contain one non-null child
   3. The number of nodes in t that contain two non-null children



1. (Source: Weiss 18.10) Suppose a binary tree stores ints. Write efficient methods (and give their Big-Oh running times) that take a reference to a binary tree root t and compute
   1. The number of even data items
   2. The sum of all the items in the tree

### C. Binary search trees

1. (Source: Weiss 19.1) Show the result of inserting 3, 1, 4, 6, 9, 2, 5, and 7 in an initially empty binary search tree. Then show the result of deleting the root.
2. (Source: Weiss 19.2) Draw all binary search trees that can result from inserting permutations of 1, 2, 3, and 4. How many trees are there? What are the probabilities of each tree’s occurring if all permutations are equally likely?
3. (Source: Weiss 19.15) Implement the BinaryTree methods find, findMin, and findMax recursively.

## D. Maps

Use the MapDemo.java class from the code bundle, the Oracle online Java documentation (at [**https://docs.oracle.com/javase/8/docs/api/java/util/Map.html**](https://docs.oracle.com/javase/8/docs/api/java/util/Map.html)) and your preferred IDE (e.g. Eclipse) to work out answers to the following questions about maps.

1. How do you enter a new pair into a map object?
2. Maps have at most one value per key. What happens if you enter a pair with a key that already exists in the map?
3. How do you find all the keys in a map (the map’s domain)?
4. How do you find all the values in a map (the map’s range)?
5. Describe two ways to remove a pair from a map.

## E. Collections API

1. Consider the four core interfaces of the Collections API: Set, List, Queue, Map. For each of the four assignments below, specify which of the four core interfaces is best-suited to the problem, and explain how to use an implementation of it to implement the assignment. You can complete the code for this in CollectionsDemo.java.
   1. Whimsical Toys Inc (WTI) needs to record the names of all its employees. Every month, an employee will be chosen at random from these records to receive a free toy.
   2. WTI has decided that each new product will be named after an employee but only first names will be used, and each name will be used only once. Prepare a list of unique first names.
   3. WTI decides that it only wants to use the most popular names for its toys. Count up the number of employees who have each first name.
   4. WTI acquires season tickets for the local lacrosse team, to be shared by employees. Create a waiting list for this popular sport.