Practice Problems 4/18/2016

Part 1: Binary Search

Using the methods below for binary search, and the array int[] list = [10 20 30 40], answer the following questions.

```
public int find(T target, T[] list) {
     return binarySearch(target, events, 0, list.length);
}

private int binarySearch(T target, T[] list, int start, int end) {
     int mid = (start + end) / 2;
     if (target.compareTo(list[mid]) == 0) {
          return mid;
     }
     if (end - start == 1) {
          return target.compareTo(list[mid]) < 0 ? start : end;
     }
     if (target.compareTo(events[mid]) < 0) {
          return binarySearch(target, list, start, mid);
     }
     if (target.compareTo(events[mid]) > 0) {
          return binarySearch(target, list, mid, end);
     }
     return 1;
}
```

1. What does find(10, list) return?

2. What does find(30, list) return?

3. What does find(5, list) return?

4. What does find(50, list) return?

5. If the values of mid were printed, what would the console display when find(10, list) is called?

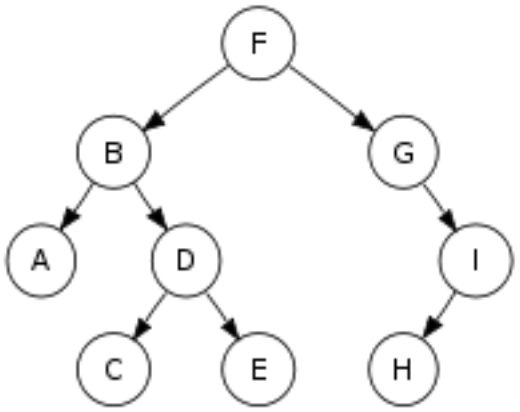
6. If the values of mid were printed, what would the console display when find(30, list) is called?

7. If the values of mid were printed, what would the console display when find(5, list) is called?

8. If the values of mid were printed, what would the console display when find(50, list) is called?

Part 2: Trees

Use the tree below to answer the following questions



1. What is the height of the tree?

2. Which nodes have leaves?

3. Which node is the root?

4. What needs to be added to make this tree full?

5. What needs to be added to make this tree complete?

6. What is the inorder traversal of this tree?

7. What is the preorder traversal of this tree?

8. What is the postorder traversal of this tree?

Part 3: Java basics

1. What is the difference between an abstract class and an interface? (very common interview question)

2. Do you extend or implement an abstract class?

3. Do you extend or implement an interface?

4. Write a generic method to exchange the positions of two different elements in an array.

```
public swap(T[] a, int i, int j)
{


}
```