CMSC 132 Quiz 4 Worksheet

The fourth quiz for the course will be on Thursday, October 13. The following list provides more information about the quiz:

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible. **You must use pencil.**
- Check the information available at http://www.cs.umd.edu/~nelson/classes/examRules.html

The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

1. What is the Java Hash Code Contract?
2. Is the Java Hash Code contract satisfied by a class that does not overload any methods of the Object class?
3. Is it incorrect for two object to have the same hashCode() value? Is there any advantage of having unique hashCode() values?
4. What is the big O associated with a perfect hash function?
5. What's the difference between a LinkedHashSet and a HashSet?
6. What's the difference between a HashMap and a TreeMap?
7. Does the TreeMap class implement the Iterable interface?
8. What is the critical section of an algorithm?
9. Calculate the asymptotic complexity of the code snippets below (using big-O notation) with respect to the problem size n. Identify the critical section(s) of each snippet.

a. f(n) = O( )

```
int a=1, b;
while (a <= n/2) {
        b = 1;
        while (b <= 2*n) {
                System.out.println(a * b);
                b++;
        }
        a++;
}
```

b. f(n) = O( )

```
for (int i=0; i<=n/2; i += n/2) {
        for (int k=0; k<n/2; k++) {
                System.out.println(i + k);
        }
}
```

c. f(n) = O( )

```
for (int k=0; k<n-1; k++) {
        for (int m=1; m<n; m*=2)
                System.out.println(k*m);
}
```

10. The **Assistants** class maintains, for a set of courses, the set of TAs for each course.

```
public class Assistants {
    Map<String, Set<String>> map;

    public Assistants() { // IMPLEMENT METHOD }
    public void addTA(String course, String taName) { // IMPLEMENT METHOD }
    public void displayTAsPerCourse() { // IMPLEMENT THIS METHOD      }
}
```

What You Must Implement

1. Implement a constructor for Assistants that creates an empty map.
2. Implement the **addTA** method that adds a teaching assistant to a specific course. A map entry for the course must be created if one does not exist.
3. Implement the **displayTAsPerCourse** method that prints (using System.out.println) the name of a course followed by the TAs for the course.

11. The class **MapquestJr** uses a Java map (`Map<List<String>,String>`) to keep track of information about roads between cities. For example, to represent that a road named "I-98" exists between city "A" and "B" we create a list with elements "A" and "B", and then map that list to the string "I-98". An incomplete MapquestJr class is presented below.

```
public class MapquestJr {
    Map<List<String>,String> roadInfo;

    public MapquestJr() {
        // IMPLEMENT METHOD
        roadInfo =
    }

    public void addRoadInfo(String startCity, String endCity, String name) {
        ArrayList<String> entry = new ArrayList<String>();
        // IMPLEMENT METHOD
    }

    public List<String> getRoute(List<String> cities) {
        List<String> route = new ArrayList<String>();
        // IMPLEMENT METHOD
        return route;
    }
}
```

What you must implement

1. Complete the above constructor so an appropriate map object is created.
2. Complete the **addRoadInfo** method. This method creates an entry in the map that stores the name of the road that exists between the cities represented by startCity and endCity. For simplicity you can assume there is only one road between any two cities.
3. Complete the **getRoute** method. This method returns a list with the names of roads we must follow in order to visit the cities provided in the parameter. The cities should be visited in the same order they appear in the parameter. You can assume that the parameter represents a route for which there are roads connecting the specified cities.