CSSE221: Fundamentals of Software Development Honors

Homework 6 Solutions

Programming assignment

Finish Markov part 1, as stated in the specification, being sure you have checked in your final, fully documented copy to the repository.

Written homework to do before week 6 (due before beginning of class next Monday).

As usual, please bring hardcopy for written answers to class to hand in on Monday.

1. No late days for questions 3-7, so I can post solutions in time to study for the exam. However, you may use a late day if you need it for Q8.

2. Make sure you are familiar with the material in the reading for the week.

3. Write a method that takes a linked list as an argument and removes every other element from it. It should mutate the list, so will be a void function. You can verify your answer by writing the code (10 pts)

   SOLUTION:

```
static void removeEveryOther(LinkedList<String> list) {
        for (Iterator<String> iter = list.iterator(); iter.hasNext(); iter.next()) {
                iter.next();
                iter.remove();
        }
}
```

   Note: This does not have to be a list of Strings. It can be a list of any type of elements.

4. Say we want to sort a list of n items. Which of the data structures that we studied could you insert all n items into and then remove them in order and have the data come out sorted, and it take O(n log n) total time? Explain your answer. (5 pts)

   SOLUTION:

   We could use a HashSet. The elements are removed in sorted order and Insertion and removal takes O(log n) for each of n items.

5. Insert, in this order, the Strings "exam", "two", "tuesday" into a HashSet first, then a TreeSet, and then print each set (each class has a toString() method). Show the outputs and explain they are different (4 pts)

SOLUTION:

[two, tuesday, exam]
[exam, tuesday, two]

The HashSet output is neither sorted, nor in the order it was input, but in the order that the hashcodes of the inputs appear. The TreeSet output is sorted.

6. Complete the **Team Preferences for Simulation Project SECTION X** survey on ANGEL, where X is your section number. The survey **MUST be completed by this THURSDAY** so I can assign you to your project teams before next Monday. The survey is accessible following these links on the course ANGEL page: Lessons → Miscellaneous → Team Preferences for Simulation Project SECTION X.

SOLUTION:

Done on ANGEL

7. Brainstorm 3 ideas for the Simulation project that interest you. Write them on separate paper and bring them to class (6 pts)

SOLUTION:

Answers will vary

8. Finish the recursion exercises you started in class:

   a. How often are the **Hofstadter Female and Male Sequences** in the slides different in the first 50 positions? first 500? first 5,000? first 5,000,000? (10 pts)

      i. Write your four answers as comments at the top of the file containing the code you used to solve this.
      SOLUTION: (in Students' repos)

First 50:	**8**
First 500:	**13**
First 5,000:	**18**
First 5,000,000:	**32**

      ii. Commit your work!

b. Complete the recursive **drawSierpinski()** method in the **SierpinskiRenderer** class in the **sierpinski** package. This method should render the Sierpiński Triangle as shown in the figure below. The triangle is rendered by following these steps: (25 pts)

i. Draw a solid equilateral triangle.
ii. In a contrasting color, draw another solid equilateral triangle whose corner points are the midpoints of the original's sides.
iii. Repeat this process recursively for each of the three corner triangles. That is, you will need **three** recursive calls in your method.
iv. Technically speaking, this process is repeated an infinite number of times to create the true Sierpiński triangle, but we don't have that much time. So, stop your recursion when the length of a side of the triangle becomes shorter than some fixed constant, say `5`.

SOLUTION: (in Students' repos)