CSSE 221: Fundamentals of Software Development Honors

Key

Programming assignment (due by Saturday Oct 20, 11:59 pm)

You should have checked in the version of your Simulation project corresponding to where you are on your schedule.

Written homework due before beginning of class on *Thursday*, Oct 18).

1. Complete all the reading.
2. (12 pts) Weiss 6.4

   SOLUTION:

   TreeSet, PriorityQueue

3. (8 pts) Weiss 6.6

   SOLUTION:

   Run time = O(1)

4. (10 pts) Weiss 6.12 (a,b only); assume that the elements are sorted in increasing order.
   a. SOLUTION:

      findMin – returns a reference to the first item in the array
      deleteMin – returns a reference to the first item in the array, shift all items forward
      insert – iterate to the location in the array where the element belongs, shift all other elements toward the back and then store the value into the array

   b. SOLUTION:

      findMin – O(1)
      deleteMin – O(N)
      insert – O(N)

5. (10 pts) Weiss 6.13 (a,b only)
   a. SOLUTION:

      findMin – loops through the entire list, storing a reference to the smallest value until it reaches the end, then returns a reference to the smallest
      deleteMin – loops through to find the min, then removes the element and shifts all forward, then returns the reference to the minimum
      insert – add to the end of the array

   b. SOLUTION:

      findMin – O(N)
      deleteMin – O(N)
      insert – O(1)

6. (10 pts) Weiss 6.14 (a,b only)
    a. SOLUTION:
        insert – inserts at the end of the array
        findMin – returns a reference to the element based on the additional data member
        deleteMin – removes from the known minimum location, shifts all forward, loops to the new minimum

    b. SOLUTION:
        findMin – O(1)
        deleteMin – O(N)
        insert – O(1)

7. (10 pts) Weiss 6.21

SOLUTION:

```
private static void removeEveryOther(List<Integer> list) {
      Iterator<Integer> itr = list.iterator();
      boolean toggle = true;
      while (itr.hasNext()) {
            itr.next();
            if (toggle) { itr.remove(); }
            toggle = !toggle;
      }
}
```

You **need** to use an iterator to do this one, since List's remove(i) is O(n), and you are doing n/2 removes, giving $O(n^2)$ runtime, which isn't allowed in the problem.

An interesting related problem is how to do this if it's a definitely a LinkedList and you have access to its internal structure.
Grading: if iterator, gets >=5 pts. If use set/get, this is $O(n^2)$, so -5..

8. (20 pts) Please finish the Sierpinski Gasket project and check in your code to your personal repository.
9. (20 pts) Please do the Fibonacci efficiency exercise stated here, making sure your code is in your personal repository as well.