CSE12 - Lecture 23 - B00

Monday, November 21, 2022   9:00 AM

PA8 due tomorrow

PA5 Late/Resubmit due tomorrow

Holiday ⇒ Thur/Friday

PA7/PA8 Late/Resubmit ⇒ due Friday of Week 10

Create a RandomStream class that generates random numbers in an enhanced for loop:.

Generating a random number:

From java.util.*

→ class Random
       public int nextInt(int bound)

       Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value
       (exclusive), drawn from this random number generator's sequence.

Random random;
int value = random.nextInt(100); //random number between 0 and 99

Note: always use a field, never create a new one over and over again in a loop.

```
RandomStream r = new RandomStream(10, 100);
for (Integer i : r) {
   System.out.println(i);
 }
```

```
class                implements Iterable<Integer>
public RandomStream(int size, int bound) {
    int bound;
    int size;
    Random number;
    public RandomStream(int size, int bound) {
        this.size = size;
        this.bound = bound;
        number = new Random();
    }
    class RandomIterator implements Iterator<Integer> {
        int current = 0;

        public boolean hasNext() {
            return this.current < size;
        }

        public Integer next() {
            int value = number.nextInt(bound);
            this.current += 1;
            return value;
        }
    }
    public Iterator<Integer> iterator() {
        return new RandomIterator();
    }
}
```
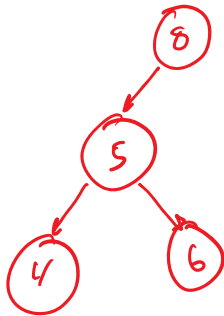
55
75
61
0
1
99 → bound-1
7
10
11
30

How would we make a BST iterator?

pre - traversal
post -
in- order -

Iterator class
└ save state
    └ create an array list
    └ fill AL w/ an inorder traversal

    └→ iterator → next/hasNext same as
                              Array List

8
5
4
6

How would we make a Heap iterator?

① copy the heap into the iterator
    └ use the poll() of the copy in next
        └ all elements in heap order

② copy the array into the iterator
    └ sort in heap order
    └ in next() just use the array
        like an arraylist version of iterator