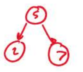
Exam 3 tomorrow 8 am → due Sun 8am

All PA & LQ → due Next Friday @ 1pm
↳ Form → shut down early next week
↳ every thing submitted graded

Review

How would we make a BST iterator?



5 2 7
pre-order traversal

2 5 7
in-order traversal

Iterator
↳ state ——→ traverse BST
↳ more & next ↳ save in an array list (either keys only or key/value pair)

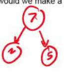
2	5	7

Data structure should not change during iteration
↳ Java → throws exception

step

Max

How would we make a Heap iterator?



7	2	5

→ max heap order 7, 5, 2

Iterator
↳ state
↳ new heap → same data
↳ next() → poll() on internal heap

class Heap {
class Inner Heap {
Heap < >
next() poll
}
}

Complete expandCapacity for the Circular ArrayList

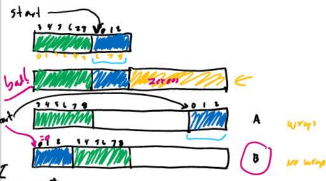
```
public class CAList<E> implements List<E> {
  E[] contents;
  int size;
  int start;

  @SuppressWarnings("unchecked")
  public CAList(int capacity) {
    this.contents = (E[])(new Object[capacity]);
    this.size = 0;
    this.start = 0;
  }

  private int indexFor(int index) {
    int ans = (this.start + index) %
              this.contents.length;
    System.out.println("Index for " + index +
                       " is " + ans);
    return ans;
  }

  public E get(int index) {
    // ASSUME index is in bounds
    int toLookup = this.indexFor(index);
    return this.contents[toLookup];
  }

  public void prepend(E value) {
    expandCapacity();
    this.size += 1;
    this.start -= 1;
    if (this.start == -1) {
      this.start = this.contents.length - 1;
    }
    this.contents[this.start] = value;
  }

 public static void main(String[] args) {
    CAList<Integer> a = new CAList<>(10);   // 10 crossed out, 2
    System.out.println(a);
    a.prepend(30);
    System.out.println(a);
    a.add(40);
    System.out.println(a);
    a.prepend(20);
    System.out.println(a);
    a.add(70);
    System.out.println(a);
  }
```

1	0
40	30

1	2		0
30	40		20

1	2	3	0
30	40	70	20

```
@SuppressWarnings("unchecked")
private void expandCapacity() {
  int currentCapacity = this.contents.length;
  if(this.size < currentCapacity) { return; }

  // save old array
  E[] oldArray = this.contents;

  // create new array
  E[] newArray = (E[]) new Object[currentCapacity * 2];

  // iterate through old array
  for (int i=0; i<this.size; i++) {
    //newArray[i] = oldArray[i];
    newArray[i] = oldArray[this.indexFor(i)];
  }

  // copy contents into our field
  this.contents = newArray;

  // reset start
  this.start = 0;
}
}
```



Write several tests to confirm that expandCapacity works

What is the run-time of prepend()?

Worst Case $\Theta(1) + \Theta(n)$ (ec)

Best Case $\Theta(1)$

Average Case $\Theta(1)$ per prepend
↳ amortized analysis

How would implement remove on a Circular ArrayList?

start

2	3	0	1

2	3		0

shiftUp()

3		0	1

shiftDown()

How would implement insert on a Circular ArrayList?