Ravi Sinha

October 10, 2016

Homework 2 Written

Problem 1:
You are given a list, L , and a another list, P, containing integers sorted in ascending order. The operation printLots(L,P) will print the elements in L that are in positions specified by P. For instance, if P=1,3,4,6, the elements in positions 1,3,4, and 6 in L are printed. Write the procedure printLots(L,P).

```
public static void printLots(ArrayList<Integer> L, ArrayList<Integer> P) {
	Iterator<Integer> itrP = P.iterator();
	Iterator<Integer> itrL = L.iterator();

	int count = -1;
	int element = 500;
	int position;

	if (P.isEmpty()) {
	System.out.println("List P is Empty");
	}

	if (L.isEmpty()) {
	System.out.println("List L is Empty");
	}


	while (itrP.hasNext()) {
	position = itrP.next();
	while (count != position) {
		count++;
		if (itrL.hasNext()) {
			element = itrL.next(); }
		else {
			System.out.println("Error: Out of Bound");
		break;
		}
	}
	if (itrL.hasNext()) {
		System.out.println("The integer in the position is: " + element);

	}
}

}
```

Problem 2 (10 pts): Weiss 3.4

Given two sorted lists, L_1 and L_2, write a procedure to compute $L_1 \cap L_2$ using only the basic list operations.

1) Given two sorted lists L1 and L2
2) Create new list (call it result)
3) Set nodes of L1 and L2 equal to the head
4) While loop: L1 node and L2 node !=NULL, run below
    a. If L1 node's value is smaller than L2 Node, set L1 node to its next node
    b. If L1 node's value is greater than L2 Node, set L2 node to its next node
    c. If L1 and L2 node value are the same, then add the value to the created list result

Problem 3 (10 pts): Weiss 3.24

Write routines to implement two stacks in one array.

```
public class TwoStacks{

    public TwoStacks(int size){
            //when popping, only the counter changes.
            //the stack's TOP is going to be in the middle of the array
            total = size of stack;
            //will decrement when an element is pushed and increment when popped
            //start both of these outside array indices so we can monitor empty stacks
            leftCounter = -1;
            rightCounter = size;
    }

    public AnyType pushLeft(AnyType a){
            //add elements a from the leftcounter starting at first position
            if(total > 0)
                    leftCounter++;
                    stack at position[leftCounter] = a;
                    total--;
            return a;
    }

    public AnyType popLeft(AnyType a){


            if(leftCounter < 0){
                    empty stack
            }else{
                    leftCounter--;
                    total++;
            }
    }
```

```
    public AnyType pushRight(AnyType a){
        if(total != 0)
            //push elements a into stack from the right side
            rightCounter--;
            stack at position[rightCounter] = a;
            total--;
        return a;
    }

    public AnyType popRight(AnyType a){
        if(rightCounter >= end of stack)
            empty stack
        else
            rightCounter++;
            total++;

    }

}
```

1. Problem 4 (10 pts):
   Answer the following questions:
   - (a) Provide a solution for this specific input train and 3 holding tracks as a sequence of steps.
     1. Move car #4 to holding track S1
     2. Move car #3 to holding track S2
     3. Move car #1 to back of the output track
     4. Move car #8 to holding track S3
     5. Move car #2 to back of output track
     6. Move car #3 from S2 to output track
     7. Move car #4 from S1 to output track
     8. Move car #7 to holding track S3 (on top of car #8)
     9. Move car #6 to holding track S2
     10. Move car #9 to holding track S1
     11. Move car #5 from input track to output track
     12. Then remove car #6 from holding track S2 and move it to output track
     13. Then remove car #7 from holding track S3 and move it to output track
     14. Then remove car #8 from holding track S3 and move it to output track
     15. Then remove car #9 from holding track S1 and move it to output track

     Output track should now read: 9 8 7 6 5 4 3 2 1

   - (b) Show an example for a train of length 9 that cannot be rearranged in increasing order using 3 holding tracks.
     - 5,4,8,1,7 ,9,6,3, 2