Matthew Barlow Written Homework 2

**Problem 1:**



**Problem 2:**

**[3.8] a) Why is theSize saved prior to entering the for loop?**

It is saved before the for loop because it is a parameter for the loop. As such it must be declared outside.

**b) What is the running time of removeFirstHalf if list is an ArrayList?**

O(N^2). Each .remove from the beginning requires moving all elements forward, which takes linear time.

**c) What is the running time of removeFirstHalf if list is a LinkedList?**

O(N) since remove can be done in O(1) time.

**d) Does using an iterator make removeHalf faster for either type of List?**

Yes, the Collection’s remove method must first find the item to remove, while the iterator removes the last item returned by next. It is faster to remove an item if you already "know" where it is located instead of searching for it.

**Problem 3:**

[3.24] We create two stacks from opposite sides of an array arr[]. The first stack starts at 0 and grows up towards n, while the second starts form n and grows down towards 0. Stack overflow happens when an element is pushed when the two stack pointers are adjacent. Both stacks grow (or shrink) in opposite direction.



2Stack(int a){

int arr[] = new int[a];

size = a;

int top1 = -1;

int top2 = size;

}

------------------------------------

push and pop elements onto stack1

-------------------------------------

void push1(int n){

if (top1 < top2 - 1){

top1++; arr[top1] = n;

}

else{

System.out.println("Stack Overflow"); break;

}

}

int pop1(){

if (top1 >= 0){

int n = arr[top1];

top1--;

return n;

}

return 0;

}

------------------------------------

push and pop elements onto stack2

------------------------------------- void push2(int n){ if (top1 < top2 -1){ top2--; arr[top2] = n; } else{ System.out.println("Stack Overflow"); break; }} int pop2(){ if(top2 < size){ int n =arr[top2]; top2++; return n; } return 0; } ----------- RUN EXAMPLE ----------- public static void main(String args[]) { 2Stack A = new 2Stack(5); ts.push1(1); ts.push2(2); ts.push1(3); ts.push2(4); ts.push3(5); System.out.println("Popped element from stack1 is "+ A.pop1()); System.out.println("Popped element from stack2 is " + A.pop2()); } }

**Problem 4:**

**a.)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input Track** | **S1** | **S2** | **S3** | **Output Track** |
| 596728134 |  |  |  |  |
| 59672813 | 4 |  |  |  |
| 5967281 | 34 |  |  |  |
| 596728 | 34 |  |  | 1 |
| 59672 | 34 | 8 |  | 1 |
| 5967 | 34 | 8 |  | 21 |
| 596 | 34 | 78 |  | 21 |
| 596 | 4 | 78 |  | 321 |
| 596 |  | 78 |  | 4321 |
| 59 |  | 678 |  | 4321 |
| 5 | 9 | 678 |  | 4321 |
|  |  | 678 |  | 54321 |
|  | 9 | 78 |  | 654321 |
|  | 9 | 8 |  | 7654321 |
|  | 9 |  |  | 87654321 |
|  |  |  |  | 987654321 |

**b.)** 123687954