CS61B Extra Problems 3

First, a Note:

This worksheet was stolen from Evan Ye, TA for CS 61B in Spring 2014. Thank him if you have the chance! All I've done is adapt it a bit for Spring 2016.

The class definitions are also a bit different for this worksheet than what you saw in class. Here they are. Note the **lack** of a sentinel node!!

```java
public class SListNode {
  public Object item;
  public SListNode next;
}
```

```java
public class SList {
  private SListNode head;              // First node in list.
  private int size;                    // Number of items in list.

  public SList() {                     // Here's how to represent an empty list.
    head = null;
    size = 0;
  }

  public void insertFront(Object item) {
    head = new SListNode(item, head);
    size++;
  }
}
```

Good luck on the next few problems. This is a very hard, dense worksheet, so don't be discouraged if you get stuck!

Linked Lists vs Arrays

List the characteristics of Arrays and Linked Lists. How are they the same? How are they different?

Box and Pointer

Draw a box and pointer diagram for the following SList:

```
SList list = new SList();
list.insertFront("Hello!");
list.insertFront("Sup?");
list.insertFront("First!");
```

Head Spinning a la Hilfinger

```
int[][] magic = { { 1, 5, 9 }, { 6, 7, 2 }, { 8, 3, 4 } };
int[][] m2 = magic;
int[][] m3 = new int[3][];
for (int i = 0; i < 3; i += 1) m3[i] = magic[i];
m3[2] = null; m2[1] = null; magic[0][1] = 2;
```

What are magic[2], magic[1], m2[2], m3[1], m2[0][1], m3[0][1] equal to?

After, I set magic = null. Now what is m2[0][1] ?

Hint: Draw a box and pointer if you can't figure it out.

Mystery

What does the following code segment do?

```
void mystery(int[] arr) {
  if (arr.length == 0)
    return;
  int tmp = arr[arr.length-1];
  for (int i = arr.length-2; i >= 0; i -= 1)   // Why backwards?
    arr[i+1] = arr[i];
  arr[0] = tmp;
}
```

Even more Linked Lists!

Write a removeFront method for an the SList class given (again, note that it is slightly different from what we saw in lecture!) Maintain the invariant that the head and size are correct.

Spring 2009 MT1 Question 3a

Write a method called removeNode for the SList class given on page 1. removeNode takes an SListNode node which you know is in this list, and removes it. (Do not worry if the node does not happen to be in the list.)