CS 61B Discussion 3 Spring 2015

1 Array Insertion

Write a method that inserts val into the given position in x. For example, if x = [5, 9, 14, 15], val = 6, and position = 2, then the method should return [5, 9, 6, 14, 15]. You may assume the position is valid.

```
public static int[] insert(int[] x, int val, int position) {

        int[] returnArray = new int[x.length+1];
        int returnArrayPosition = 0, xPosition = 0;
        while (returnArrayPosition < returnArray.length) {
                if (returnArrayPosition == position) {
                        returnArray[returnArrayPosition] = val;
                } else {
                        returnArray[returnArrayPosition] =
                            x[xPosition];
                        xPosition++
                }
                returnArrayPosition++;
        }
        return returnArray;

}
```

Is it possible to write a version of this method that returns void and changes x in place (i.e. destructively)? No, arrays are of fixed size so you must reallocate space for a new array to insert an extra element.

2 Singly Linked Lists

For the following problems, use the following implementation of an SNode:

```
public class SNode {
        public SNode next;
        public double val;
        public SNode(double val, SNode next) {
                this.next = next;
                this.val = val;
        }
}
```

Given the following structure for a singly linked list, write a method to insert elements into the given position. If the position is invalid, insert the new node at the end of the list. For example, if the SList is 5 –> 6 –> 2, insert(10, 1) would result in 5 –> 10 –> 6 –> 2.

```
public class SList {
        private SNode head;
        public void insert(double val, int position) {

                if (head == null || position == 0) {
                        head = new SNode(val, head);
                } else {
                        SNode cur = head;
                        while (position > 1 && cur.next != null) {
                                position--;
                                cur = cur.next;
                        }
                        SNode temp = cur.next;
                        cur.next = new SNode(val, temp);

                }
        }
}
```

3 Sentinel Nodes

Given the the following structure for a singly linked list using sentinel nodes, write a method to insert elements into it. If the position is invalid, insert the new node at the end of the list.

```
public class SentinelSList {
        private SNode front;
        private SNode back;
        public SentinelSList() {
                this.back = new SNode(0, null);
                this.front = new SNode(0, back);
        }
        public void insert(double val, int position) {

                SNode cur = front;
                int curPos = 0;
                while (curPos < position && cur.next != back) {
                        cur = cur.next;
                        curPos++;
                }
                SNode temp = cur.next;
                cur.next = new SNode(val, temp);

        }
}
```

Challenge Problem: Write a method `xify(int[] x)` that replaces the ith number with x[i] copies of itself. For example, `xify([3, 2, 1])` would return [3, 3, 3, 2, 2, 1].

```java
public static int[] xify(int[] x) {
        int total = 0;
        int i = 0;
        while (i < x.length) {
                total+= x[i];
        }
        int[] newArr = new int[total];
        int count = 0;
        i = 0;
        while (i < x.length) {
                int j = 0;
                while (j < x[i]) {
                        newArr[count] = x[i];
                        count++;
                        j++;
                }
                i++;
        }
        return newArr;
}
```