Estructuras de Datos **Nombre:** Santiago Baldrich
2014 - I
Parcial 1
22/04/14
Tiempo límite: 1 hora y 45 minutos

Este examen contiene 3 paginas y 4 problemas.

No está permitido el uso de libros, apuntes ni dispositivos electrónicos para la presentación de este examen.

Tenga en cuenta las siguientes recomendaciones:

- **Apague cualquier dispositivo electrónico** mientras se encuentre presentando el examen.
- **Organice su trabajo** de forma coherente, las respuestas presentadas de forma desorganizada serán penalizadas fuertemente.
- **Las respuestas sin sustentación no serán consideradas como completas.** Una respuesta correcta que no contenga los cálculos o explicaciones debidas no recibirá una calificación completa.
- **Cualquier intento de fraude será tratado según la normativa vigente de la universidad.**
- No escriba en la tabla de la derecha.
- La interpretación del examen hace parte del examen.

Problema	Puntos	Calificación
1	10	112639876
2	20	90000000!!
3	10	1e(9e1000)
4	10	1/0
Total:	50	Inf

Valar morghulis.

---

1. (10 puntos) Dada la clase `ArrayLinearListImproved` vista en clase, cree un método `merge` que reciba otra lista y agregue sus elementos a ésta. No puede usar el método `add` definido previamente en `ArrayLinearList`. En el repositorio

2. (20 puntos) Defina una clase `SortedChain` que herede de la clase `LinkedList` vista en clase. `SortedChain` se caracteriza porque sus elementos se mantienen ordenados siempre. escriba la declaración de la clase y los métodos `add`: *recibe un elemento y lo agrega en la posición que*

le corresponde manteniendo el orden de la cadena y `remove`: *recibe un entero que indica la posición del elemento que se quiere eliminar.* En el repositorio

3. (10 puntos)¿Cuál es la complejidad de tiempo de las siguientes operaciones? (Responda al frente de cada pregunta)
   (a) (2 puntos) `indexOf` en `ArrayLinearList` O(size)
   (b) (2 puntos) `add` en `LinkedList` (`Chain`) O(index)
   (c) (2 puntos) `remove` en `ArrayLinearList` O(size - index)
   (d) (2 puntos) `sort` en `ArrayLinearList` O(nlog(n))
   (e) (2 puntos) `size` en `LinkedList` (`Chain`) O(1)

4. (10 puntos) Dado el siguiente código diga si compilaría o no. En caso de que no, haga las correcciones pertinentes (sobre el mismo código, no hace falta que lo haga en la hoja examen).

```
1  package co.edu.unal.ds.list;
2
3  import java.util.*;
4
5  @SuppressWarnings("unchecked")
6  public class CustomList<T super ?> implements Iterable<T>{
7
8          protected T[] element;
9          protected int size;
10         public ArrayLinearList(){
11                 super(10);
12         }
13
14         public ArrayLinearList(int initialCapacity){
15                 element = (T[]) new T[initialCapacity];
16                 size = 0;
17         }
18
19         public boolean isEmpty(){
20                 return size==0;
21         }
22
23         public int size(){
24                 return size;
25         }
26
27         private void checkIndex(int index){
28                 if( index < 0 || index >= size )
29                         throw new IndexOutOfBoundsException();
30         }
31
32         public void add(int index, T elem){
33                 if( index < 0 || index >=size )
```

```
34                      throw new IndexOutOfBoundsException();
35
36              if( size==element.length ){
37                      T[] old = element;
38                      element = (T[]) new Object[2 * size];
39                      System.arraycopy(old, 0, element, 0, size);
40              }
41
42              for(int i = size; i>index; i--)
43                      element[i+1] = element[i];
44              ++size;
45              element[index] = elem;
46
47      }
48
49      public T get( int index ){
50              checkIndex( index );
51              return element[index];
52      }
53
54      public int indexOf( T elem ){
55              for( int i=0; i<size; ++i )
56                      if( elem == element[i] )
57                              return i;
58              return -1;
59      }
60
61      public T remove( int index ){
62              checkIndex( index );
63              T removed = element[ index ];
64              for( int i=index; i<size-1; i++)
65                      element[ i ] = element[ i+1 ];
66              element[ size-- ] = null;
67              return removed;
68      }
69
70 }
```