Programación I

Parcial (turno tarde) – 12/06/2019

Ejercicio 1 *(20 pts. - 5 pts. cada item.)*

Discutir la veracidad de las siguientes afirmaciones, justificando su respuesta.

a) Un arreglo en Java es una colección de elementos, donde no todos los elementos deben ser del mismo tipo.
b) Cuando un objeto está roto decimos que no se definió su invariante de representación.
c) Si al comparar dos objetos con el método equals() el resultado dio verdadero entonces el operador == también lo hará.
d) El algoritmo de ordenamiento QuickSort tiene orden de complejidad en peor caso $O(n \log(n))$

Ejercicio 2 *(20 pts.)*

Escribir una función recursiva String estaPrimera(String s1, String s2) que toma dos String s1 y s2 y devuelve el String que esta primero en el diccionario. Por ejemplo: Por ejemplo:

- alternados(''piedra'',''cantor'') debe devolver ''cantor''.
- alternados('''',''perro'') debe devolver ''''.
- alternados(''hielo'',''holanda'') debe devolver ''hielo''.
- alternados(''candelabro'',''canario'') debe devolver ''canario''.
- alternados(''materia'','''') debe devolver ''''.

Se pide resolver **utilizando recursión**. Se pueden dar por hecha la función String resto(String s) que devuelve una cadena igual a s pero sin su primer carácter.

Ejercicio 3 *(35 pts. - 15 pts. item a.- y 20 pts. item b)*

Consideremos las clases UNGS, Comsion, Docente y Estudiante definidas como:

```
public class UNGS
{
    Comision[] comisiones;
    ...
}
public class Comision {
    String materia;
    int numero;
    Docente[] docentes;
    Estudiante[] inscriptos;
    int[] calificaciones;
    ...
}
```

```
public class Docente {
    String nombre;
    int dni;
    ...
}

public class Estudiante {
    String nombre;
    int legajo;
    ...
}
```

Los arreglos inscriptos y calificaciones de una Comision tienen el mismo tamaño y el valor de calificaciones[i] indica la calificación obtenida por el estudiante inscriptos[i]. Para la clase UNGS, se pide:

a) Escribir un método **int** enCuantasComisiones(Estudiante e) que devuelve la cantidad de comsiones en las que está inscripto el estudiante e.
b) Se pide escribir un método ArrayList<String> faltanDocentes() que devuelva una lista con todas las materias que tienen falta de docentes. En una materia faltan docentes, si más de diez estudiantes por cada docente.
c) (*bonus track* 20 pts.) Escribir un método Estdiante mejorPromedio() que devuelve el estudiante con el mejor promedio de calificaciones de la UNGS. Si hubiese más de un estudiante en estas condiciones puede devolver cualquiera de ellos.

Ejercicio 4 *(25 pts.)*

Dadas las clases NodoInt y ListaInt cuyas variables de instancia son las siguientes:

```
public class NodoInt
{
    int elemento;
    NodoInt siguiente;
}
```

```
public class ListaInt
{
    NodoInt primero;
    ...
}
```

Se pide para esta clase escribir el método de instancia ListaInt agregarConOrden(**int** k) que genera una nueva lista que contiene a k y todos los elementos de la lista original menores a k están en la nueva lista a la izquierda de k y los mayores o iguales a la derecha. Por ejemplo:

- Si la lista es [2,7,6,0,8,1,9] y k=5 puede devolver [1,0,2,5,7,6,8,9].
- Si la lista es [1,2,3] y k=0 puede devolver [0,1,2,3].
- Si la lista es [4,3,6,1,9,2,0,2] y k=3 puede devolver [2,0,2,1,3,4,3,6,9].

Se pide además que el método implementado **sea de orden lineal**, es decir, $O(n)$ donde n es la cantidad de elementos de la lista. Justificar la complejidad del mismo.