



Ejercicios Tema 9

Ejercicios Adaptados de Apuntes y Exámenes de EDA
Germán Moltó Martínez
gmolto@dsic.upv.es
Estructuras de Datos y Algoritmos
Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Valencia

1

Implementación de Cola de Prioridad con Lista Ordenada

- ▶ Dado el modelo de ListaConPI y su implementación ofrecida por la clase LEGListaConPI se pide:
 - ▶ Desarrolla la clase ListaOrdenada como implementación del modelo de ColaPrioridad.
- ▶ Detalles de Implementación:
 - ▶ La clase ListaOrdenada tendrá un atributo de tipo ListaConPI.
 - ▶ Los elementos se mantendrán ordenados en la ListaConPI de acuerdo a su prioridad.

```
public interface ColaPrioridad<E extends Comparable<E>>{  
    void insertar(E x);  
    E eliminarMin();  
    E recuperarMin();  
    boolean esVacia();}
```

▶ 2

Uso de Cola de Prioridad en Impresión (1/2)

- ▶ Como parte de un módulo para un sistema de impresión se desea trabajar con una Cola de Prioridad de objetos de tipo TrabajoImpresion. De esta manera, los trabajos podrán ser impresos de acuerdo al siguiente esquema de prioridades:
 - ▶ Los trabajos más urgentes son aquellos con menos hojas. De esta manera, se evita colapsar el recurso con trabajos grandes.
 - ▶ A igualdad de número de hojas, se elegirá primero el TrabajoImpresion que antes fue creado.
- ▶ Se pide:
 - ▶ Diseña la clase TrabajoImpresion con el soporte necesario para sus objetos puedan ser guardados en una ColaPrioridad.

▶ 3

Uso de Cola de Prioridad en Impresión (2/2)

- ▶ Dada la siguiente interfaz:

```
public interface DriverImpresora {  
    //Almacena la nueva petición de impresión  
    void recibirTrabajoImpresion(TrabajoImpresion t);  
    //Elige el trabajo más prioritario  
    TrabajoImpresion elegirTrabajoImpresion();  
}
```

- ▶ Se pide implementar la interfaz en la clase MiDriverImpresora para que, utilizando una Cola de Prioridad, realice la gestión de los trabajos de impresión.
 - ▶ Asumir disponible la clase LPIColaPrioridad como implementación de ColaPrioridad.

▶ 4

Análisis de Secuencias de ADN

- ▶ En una aplicación para el análisis de secuencias de ADN, se desea desarrollar un módulo para ir obteniendo las secuencias de menor a mayor longitud. Para ello, se trabaja con una Cola de Prioridad.
- ▶ Se pide:
 - ▶ Modificar la clase SecuenciaADN para poder satisfacer los requisitos de la aplicación.

```
public class SecuenciaADN{  
    private String secuencia; private String descripción;  
    public SecuenciaADN(String secuencia){ this.secuencia = secuencia;}  
}
```

- ▶ Ejemplos de secuencias:
 - ▶ ATGTGAC, ATGTGTCAGAC

▶ 5

Ordenar Arrays con Cola de Prioridad

- ▶ Se pide:
 - ▶ Diseña un método **iterativo** genérico que, dado un array de elementos desordenados lo ordene ascendentemente utilizando una Cola de Prioridad.
 - ▶ Diseña un método **recursivo** genérico que, dado un array de elementos desordenados lo ordene descendentemente utilizando una Cola de Prioridad.
- ▶ Asumir que existe la clase LPIColaPrioridad como implementación de la interfaz ColaPrioridad.

▶ 6

Número de Repeticiones en una Frase

- ▶ Diseña un método estático que devuelva el número de repeticiones totales de palabras que hay en una frase dada.
 - ▶ Utilizar para ello el modelo de Diccionario y una supuesta clase ABBDiccionario que proporciona la implementación del modelo.
- ▶ Recuerda que puedes trocear una frase en palabras así:
 - ▶ `String palabras[] = frase.split(" ");`
- ▶ Ejemplos:
 - ▶ “El perro de San Roque” → 0 repeticiones
 - ▶ “Mi llama se llama Yama y enciende una llama” → 2 repeticiones
 - ▶ “Mi perro Lolo es un perro llamado Lolo” → 2 repeticiones

▶ 7

Palabras Repetidas y Frecuencia en una Frase

- ▶ Diseña un método estático que permita obtener un listado de las palabras de una frase junto a su frecuencia de repetición en una frase (o texto).
- ▶ Para ello, puedes asumir que el método toString de la supuesta clase ABBDiccionario, que implementa la interfaz Diccionario, te devuelve un listado de los pares (clave, valor).
- ▶ Recuerda que puedes trocear una frase en palabras así:
 - ▶ `String palabras[] = frase.split(" ");`
- ▶ Ejemplo:
 - ▶ la la casa es es roja
 - ▶ la 2, casa 1, es 2, roja 1

▶ 8

Traducción en Diccionario Bilingüe

- ▶ Implementar un método que reciba una frase y un diccionario bilingüe (palabra en castellano, palabra en inglés) y permita traducir la frase del castellano al inglés.
- ▶ Perfil del método:
 - ▶ `public static String traducir(String frase, Diccionario<String,String> d)`
- ▶ Recuerda que puedes trocear una frase en palabras así:
 - ▶ `String palabras[] = frase.split(" ");`