

## Primer Parcial de EDA (26 de Marzo de 2019)

1.- (2 puntos) El siguiente método de la clase **ArrayCola** invierte el orden de los elementos de una Cola, usando una **ListaConPI** auxiliar. Escribe en cada recuadro el número de la opción (ver listado a la derecha) que le corresponde.

```
public void invertir() {
```

```
    ListaConPI<E> lpi =  ();
```

```
    while ( !this.esVacia() ) {
```

```
         (  () );
```

```
         ();
```

```
    }
```

```
    while ( !lpi.esVacia() ) {
```

```
         (  () );
```

```
         ();
```

```
    }
```

```
}
```

① **this**.encolar

② lpi.insertar

③ lpi.eliminar

④ lpi.inicio

⑤ new ListaConPI<E>

⑥ new LEGListaConPI<E>

⑦ **this**.desencolar

⑧ lpi.recuperar

2.- (2 puntos) Estudia el coste Temporal del siguiente método:

```
/* Precondición: v ordenado ascendentemente AND a < b */
```

```
private static int metodo(int[] v, int ini, int fin, int a, int b) {
```

```
    if (ini > fin) { return 0; }
```

```
    int mitad = (ini + fin) / 2, res = 0;
```

```
    if (v[mitad] >= a) { res += metodo(v, mitad + 1, fin, a, b); }
```

```
    if (v[mitad] <= b) { res += metodo(v, ini, mitad - 1, a, b); }
```

```
    if (v[mitad] >= a && v[mitad] <= b) { res++; }
```

```
    return res;
```

```
}
```

a) Expresa la talla del problema **x** en función de los parámetros del método.

**x** =

(0.2 puntos)

b) Indica si hay instancias significativas para una talla dada y por qué. En caso afirmativo, descríbelas.

(0.6 puntos)

c) En base a tu apartado b), escribe la(s) Relación(es) de Recurrencia que expresa(n) el coste del método.

(0.6 puntos)

En el caso general, cuando **x** > ,

d) Resuelve la(s) Relación(es) de Recurrencia del apartado c), indicando el(los) Teoremas de Coste que usas y escribiendo el coste Temporal del método en notación asintótica (O y  $\Omega$  o bien  $\Theta$ ).

(0.6 puntos)



## ANEXO

### Las interfaces Map y ListaConPI del paquete modelos.

```
public interface Map<C, V> {  
    V insertar(C c, V v);  
    V eliminar(C c);  
    V recuperar(C c);  
    boolean esVacio();  
    int talla();  
    ListaConPI<C> claves();  
}
```

```
public interface ListaConPI<E> {  
    void insertar(E e);  
    void eliminar();  
    void inicio();  
    void siguiente();  
    void fin();  
    E recuperar();  
    boolean esFin();  
    boolean esVacia();  
    int talla();  
}
```

### Las clases NodoABB y ABB del paquete jerarquicos.

```
class NodoABB<E> {  
    E dato;  
    NodoABB<E> izq, der;  
    int talla;  
    NodoABB(E dato) {...}  
}
```

```
public class ABB<E extends Comparable<E>> {  
    protected NodoABB<E> raiz;  
    protected int talla;  
    public ABB() {...}  
    ...  
}
```

### Teoremas de coste:

**Teorema 1:**  $f(x) = a \cdot f(x - c) + b$ , con  $b \geq 1$

- si  $a=1$ ,  $f(x) \in \Theta(x)$ ;
- si  $a>1$ ,  $f(x) \in \Theta(a^{x/c})$ ;

**Teorema 2:**  $f(x) = a \cdot f(x - c) + b \cdot x + d$ , con  $b$  y  $d \geq 1$

- si  $a=1$ ,  $f(x) \in \Theta(x^2)$ ;
- si  $a>1$ ,  $f(x) \in \Theta(a^{x/c})$ ;

**Teorema 3:**  $f(x) = a \cdot f(x/c) + b$ , con  $b \geq 1$

- si  $a=1$ ,  $f(x) \in \Theta(\log_c x)$ ;
- si  $a>1$ ,  $f(x) \in \Theta(x^{\log_c a})$ ;

**Teorema 4:**  $f(x) = a \cdot f(x/c) + b \cdot x + d$ , con  $b$  y  $d \geq 1$

- si  $a<c$ ,  $f(x) \in \Theta(x)$ ;
- si  $a=c$ ,  $f(x) \in \Theta(x \cdot \log_c x)$ ;
- si  $a>c$ ,  $f(x) \in \Theta(x^{\log_c a})$ ;

### Teoremas maestros:

**Teorema para recurrencia divisora:** la solución a la ecuación  $T(x) = a \cdot T(x/b) + \Theta(x^k)$ , con  $a \geq 1$  y  $b > 1$  es:

- $T(x) \in O(x^{\log_b a})$  si  $a > b^k$ ;
- $T(x) \in O(x^k \cdot \log x)$  si  $a = b^k$ ;
- $T(x) \in O(x^k)$  si  $a < b^k$ ;

**Teorema para recurrencia sustractora:** la solución a la ecuación  $T(x) = a \cdot T(x-c) + \Theta(x^k)$  es:

- $T(x) \in \Theta(x^k)$  si  $a < 1$ ;
- $T(x) \in \Theta(x^{k+1})$  si  $a = 1$ ;
- $T(x) \in \Theta(a^{x/c})$  si  $a > 1$ ;