

Recuperación del Primer Parcial de EDA (12 de Junio de 2019)

1.- (3 puntos) En la clase **ABB**, implementa un método público que, con el menor coste temporal posible, elimine el nodo que contiene el máximo de un ABB.

Una vez diseñado el método, y suponiendo que se aplica sobre un ABB Equilibrado de N nodos y altura H , ...

a) Indica la talla del problema que resuelve (0.1 puntos)

b) Indica si hay instancias significativas para una talla dada y por qué. En caso afirmativo, descríbelas. (0.25 puntos)

c) Indica su coste Temporal utilizando la notación asintótica (O y Ω o bien Θ). (0.4 puntos)

2.- (3 puntos) Escribe un método estático **Divide y Vencerás** que, dados un array **v** de **int** ordenado ascendentemente y sin elementos repetidos y un **int x**, devuelva el elemento de **v** con valor más cercano al de **x**. Para ello...

- Puedes suponer que **v** tiene como mínimo tres elementos y que, de ellos, el elemento de valor más cercano al de **x** no está NI en la primera posición de **v** NI en la última.
- Puedes usar en tu código un método **comparar** que, como su nombre indica y en el orden de una constante, compara con **x** dos elementos **eV1** y **eV2** de **v** y devuelve aquel de ellos con valor más cercano al de **x**. Su perfil es: **int comparar(int[] v, int eV1, int eV2, int x)**.

Una vez diseñado el método, estudia su coste temporal del método recursivo que lanza. En concreto:

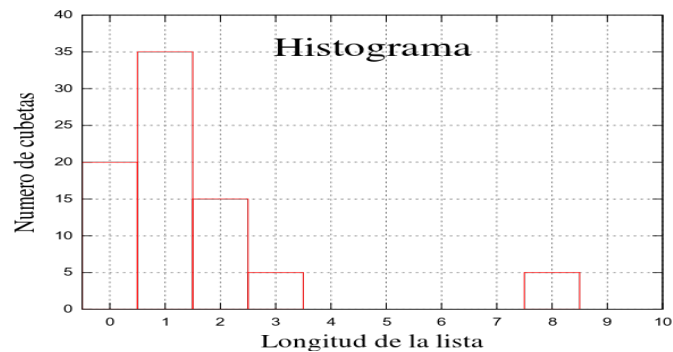
a) Expresa la talla del problema x en función de sus parámetros $x =$ (0.2 puntos)

b) Escribe la(s) Relación(es) de Recurrencia que expresa(n) su coste. (0.4 puntos)

En el caso general, cuando $x >$,

c) Resuelve la(s) Relación(es) de Recurrencia del apartado b), indicando el(los) Teoremas de Coste que usas y escribiendo el coste Temporal del método en notación asintótica (O y Ω o bien Θ). (0.4 puntos)

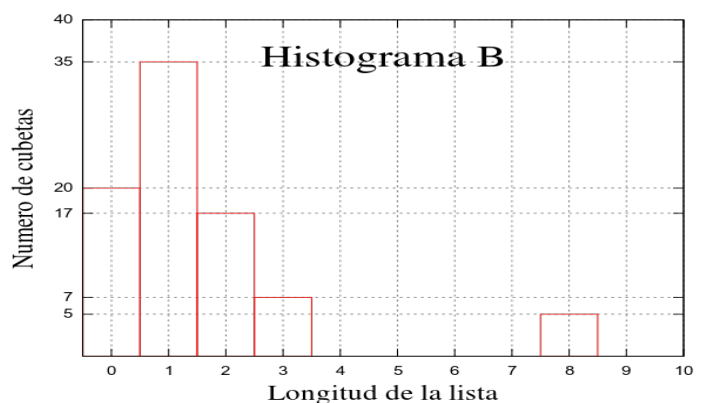
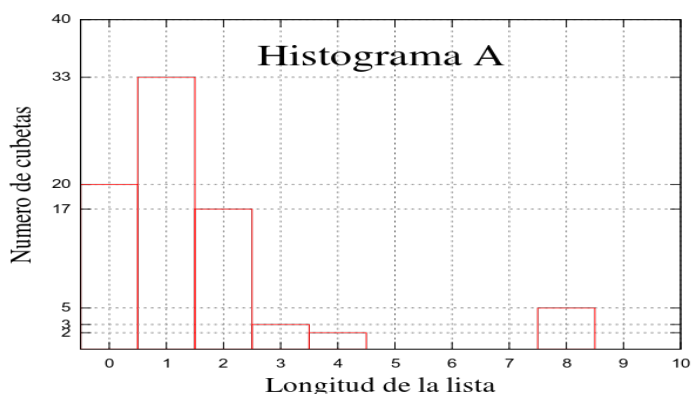
3.- (2 puntos) Dado el siguiente histograma de ocupación de una Tabla Hash (con Hashing Enlazado):



a) Indica el número de cubetas y el de elementos que tiene la Tabla, dejando indicadas las operaciones que has realizado para calcular dichos valores. (0.5 puntos)

b) Calcula el Factor de Carga de la Tabla y la desviación típica de las longitudes de sus cubetas, dejando indicadas las operaciones que has realizado para calcular dichos valores. (0.5 puntos)

c) Supón que se insertan cuatro elementos distintos y que no estaban en la Tabla. Indica cuál de los siguientes es el histograma de ocupación de la Tabla tras estas inserciones. Es imprescindible que razones tu respuesta. (1 punto)



4.- (2 puntos) El siguiente método de la clase **ArrayColaExt<E extends Comparable<E>>**, que extiende de **ArrayCola<E>**, ordena ascendentemente los elementos de una Cola usando una **ListaConPI** como estructura auxiliar. Escribe en cada recuadro el número de la opción (ver listado a la derecha) que le corresponde.

```
public void ordenar() {
```

```
     lpi = new  ();
```

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

```
        E e =  ();
```

```
        for (lpi.inicio();  &&  ;  );
```

```
             (e);
```

```
    }
```

```
    lpi.inicio();
```

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

```
         ( ());
```

```
         ();
```

```
    }
```

```
}
```

① lpi.recuperar().compareTo(e) <= 0

② this.desencolar

③ !lpi.esFin()

④ lpi.eliminar

⑤ LEGListaConPI<E>

⑥ ListaConPI<E>

⑦ lpi.insertar

⑧ lpi.siguiente()

⑨ lpi.recuperar

⑩ this.encolar

ANEXO

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 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 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 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 $a > 1$;