

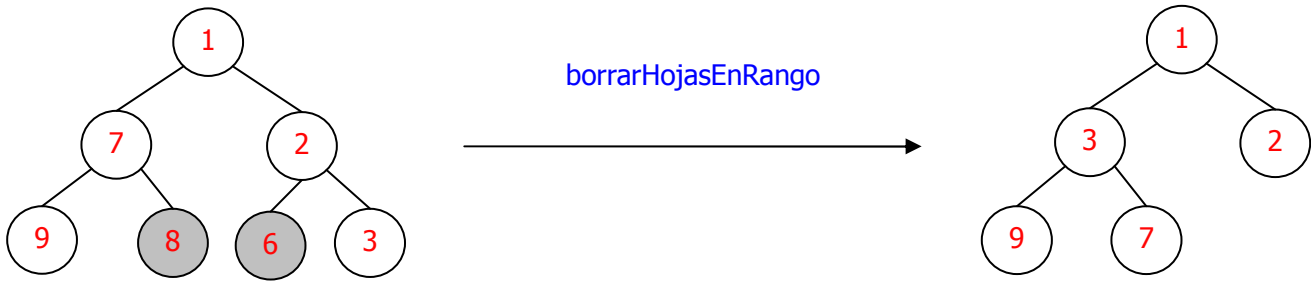


Ejercicio de seminario - ejercicio 2 borrar hojas en rango

Estructuras de datos y algoritmos (Universitat Politecnica de Valencia)

Soluciones de algunos Ejercicios de Heap

Ejercicio 2: Borrar Hojas en Rango

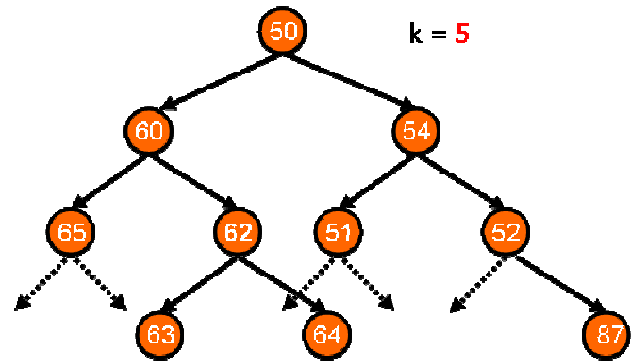
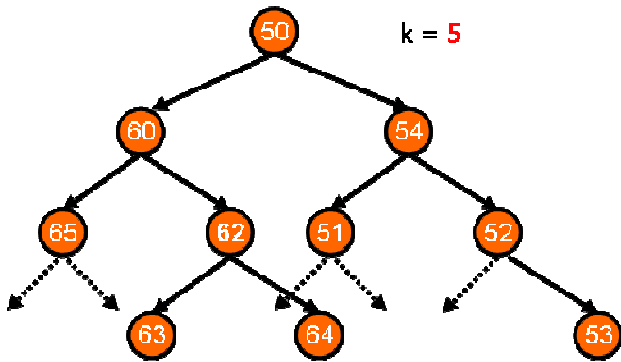


```
public void borrarHojasEnRango(E x, E y){
    // Primer paso: recorrer las Hojas del Heap borrando las que estén en [x,y]
    int i = (talla/2) + 1;
    while ( i <= talla )
        if ( elArray[i].compareTo(x) >= 0 && elArray[i].compareTo(y) <= 0 )
            elArray[i] = elArray[talla--];
        else i++;
    // Segundo paso: ¿sigue siendo Heap tras borrar las Hojas en [x,y] ?
    this.arreglar();
}
```

Ejercicio 1: Esta En

```
public int estaEn(E e){
    int res = -1;
    if ( talla != 0 ) res = estaEn(E e, 1);
    return res;
}
protected int estaEn(E e, int posActual){
    int res = -1;
    int resC = elArray[posActual].compareTo(e);
    if ( resC == 0 ) res = posActual;
    else if ( resC < 0 ){
        if ( 2*posActual <= talla ) res = estaEn(e, 2*posActual);
        if ( res==-1 && 2*posActual+1<=talla ) res = estaEn(e, 2*posActual+1);
    }
    return res;
}
```

Ejercicio 3: Eliminar K-ésimo



Primera Versión:

/** SII $1 \leq k \leq \text{tamanyo}()$: elimina el k-ésimo Dato de un Heap **/

```
public E eliminar(int k){
    E datoEnK = elArray[k]; elArray[k]= elArray[talla--];
    // ¿Sigue siendo Heap tras borrar el k-ésimo?
    this.arreglar();
    return datoEnK;
}
```

cuyo coste Temporal es $T_{\text{eliminar/arreglar}}(x=N) \in \Theta(x=N)$

Versión eficiente:

```
public E eliminar(int k){
    E datoEnK = elArray[k]; elArray[k]= elArray[talla--];
    // ¿Sigue siendo Heap tras borrar el k-ésimo?
    // reflotar elArray[k] desde k
    int posK = k; E elK = elArray[k];
    while (posK > 1 && elK.compareTo(elArray[posK/2]) < 0 ) {elArray[posK] = elArray[posK/2]; posK/= 2; }
    elArray[posK] = elK;
    // hundir elK desde posK hasta una Hoja
    hundir(posK);
    return datoEnK;
}
```

cuyo coste Temporal es como máximo $T_{\text{eliminar}}(x=N) \in O(\log x = \log N)$ y como mínimo $T_{\text{eliminar}}(x=N) \in \Omega(1)$

Ejercicio 4: Iguales Al Mínimo

```
public int igualesAlMinimo(){
    int res = 0;
    if ( talla != 0 ) res = igualesAlMinimo(1);
    return res;
}

protected int igualesAlMinimo(int posActual){
    int res = 0;
    if ( elArray[posActual].equals(elArray[1]) ){
        res++;
        if ( 2*posActual <= talla ) res += igualesAlMinimo(2*posActual);
        if ( (2*posActual)+1 <= talla ) res += igualesAlMinimo((2*posActual)+1);
    }
    return res;
}
```

Ejercicio 5: Comprobar Propiedad De Ordenación

```
protected boolean comprobarPO(int i) {  
    boolean res = true; int izqI = i*2;  
    if ( izqI <= talla )  
        if ( elArray[i].compareTo(elArray[izqI]) > 0 ) res = false;  
    else{  
        int derI = izqI+1;  
        if ( derI <= talla ){  
            if ( elArray[i].compareTo(elArray[derI]) > 0 ) res = false;  
            else res = comprobarPO(derI);  
        }  
        if ( res ) res = comprobarPO(izqI);  
    }  
    return res;  
}
```