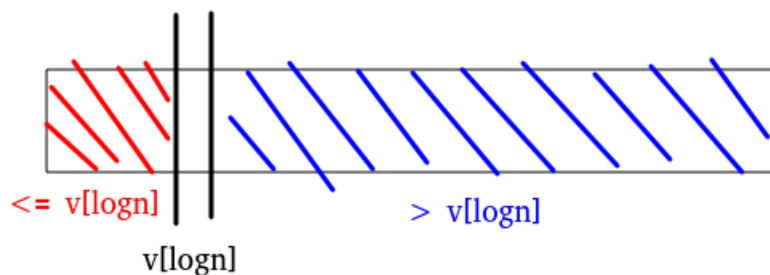


### **EJERCICIO 1.16**

Donada una taula A amb  $n$  registres, on cada registre conté un enter de valor entre 0 i  $2^n$ , i els continguts de la taula estan desordenats, dissenyeu un algorisme lineal per a obtenir una llista ordenada dels elements a A que tenen valor més gran que els  $\log n$  elements més petits a A, i al mateix temps, tenen valor més petit que els  $n - 3 \log n$  elements més grans a A.

Un posible algoritmo de coste lineal podría ser el siguiente:

1. Utilizar el algoritmo de selección determinista para encontrar el elemento  $\log(n)$ -ésimo del vector. Es importante recalcar que el algoritmo nos deja en la posición  $k$  el elemento que debería ocupar esa posición si el vector estuviera ordenado. También, a su izquierda, nos deja los elementos menores o iguales que él, y a su derecha los mayores.



2. Eliminar los elementos que quedan a la izquierda de  $v[\log(n)]$ , es decir, los menores o iguales que  $v[\log(n)]$ .

3. Volver a ejecutar el algoritmo de selección determinista para encontrar el elemento  $3\log n$ -ésimo, que deja a la derecha los  $n - 3\log(n)$  elementos más grandes del vector. Como hemos

eliminado los primeros  $\log(n)$  elementos, el elemento que queremos ordenar es el  $2\log(n)$ -ésimo.

**4.** Eliminar los elementos que quedan a la derecha de  $v[2\log(n)]$ , es decir, los mayores que  $v[2\log(n)]$ .

**5.** Nos queda ahora una lista desordenada con los elementos que cumplen la condición que nos pide el enunciado, por tanto ahora solo faltaría ordenarla. Para ello, se puede utilizar un Merge Sort, ya que en este caso estaríamos ordenando solo  $2\log(n)$  elementos.

El algoritmo tiene coste lineal, ya que el algoritmo de selección determinista en los pasos **1** y **3** se ejecuta en tiempo  $O(n)$ , para eliminar los elementos en los pasos **2** y **4** solo nos hace falta recorrer los primeros  $\log(n)$  y los últimos  $n-3\log(n)$  elementos respectivamente y para ordenar la lista tenemos coste  $O(2\log(n) \cdot \log(2\log(n))) = O(\log(n) \cdot \log(\log(n))) = O(\log(n) \cdot \log(n))$ .

Por tanto, el coste que predomina es el de la selección determinista que es  **$O(n)$** .