

**Hoja 9-3:****ANÁLISIS DE ALGORITMOS**Ejercicio 11:

Queremos obtener un listado de toda la información contenida tanto en un árbol binario de búsqueda como en una lista simplemente enlazada ordenada.

- a.- El coste temporal de la operación obtener listado es menor en el árbol binario de búsqueda, ya que en general, las operaciones en el árbol binario de búsqueda son del orden de  **$\log n$** .
- b.- El coste temporal de la operación obtener listado es menor en la lista ligada ordenada.
- c.- El coste temporal de la operación obtener listado es el mismo en los dos casos.
- d.- Es imposible obtener el listado de toda la información contenida en el árbol binario de búsqueda, ya que los procedimientos definidos sobre el árbol sólo incluyen la búsqueda, la inserción y la eliminación de elementos.

Ejercicio 12:

Entre los diferentes métodos de ordenación de vectores, en el peor de los casos:

- a.- El mejor siempre es el *Quick-Sort*, porque utiliza la técnica de divide y vencerás y su coste es logarítmico.
- b.- El mejor es el *Heap-Sort*, que en el peor de los casos sigue teniendo un coste ' $n \cdot \lg(n)$ '.
- c.- Depende del número de elementos. Si ' $n$ ' es pequeño, el mejor algoritmo es el de inserción.
- d.- El *Heap-Sort* no se puede emplear para ordenar vectores. Sólo puede ordenar montículos.

Ejercicio 13:

Supongamos que deseamos ordenar parcialmente un vector de 1000 elementos (calcular, por ejemplo, los 10 números más grandes del vector). ¿Qué algoritmo de ordenación de los vistos en clase, sería conveniente modificar para realizar esta ordenación?

Ejercicio 14:

Deseamos encontrar la mediana de un vector. Para ello nos basta con tener la mitad de sus elementos ordenados. ¿Qué método de ordenación sería conveniente modificar para determinar la mediana?

- a.- El de selección.    b.- El de inserción.    c.- El Quick-Sort.    d.- El Heap-Sort.

Razona brevemente la respuesta: