Estructura de búsqueda, inserción y eliminación:

Array

* Para hacer la búsqueda de un elemento, tendremos que empezar desde la primera posición hasta el final, en el peor de los casos (O(n))
* Para insertar elementos, solo necesitamos saber la última posición para poder colocarlo en la siguiente, siempre va a ser de orden constante (O(1))
* Para eliminar un elemento, tendremos que mover todos los elementos de la derecha a este, una posición a la izquierda. Por lo que en el peor de los casos tendremos que mover todos los elementos (O(n))

Linked list

* Para hacer la búsqueda de un elemento, pasaría igual que en el array, por lo que el recorreremos todos los elementos (O(n))
* Para insertar elementos, solo necesitamos la posición, por lo que el orden es constante (O(n))
* Para eliminar un elemento, recorremos la lista entera buscando el elemento para poder eliminarlo (O(n))

Para reducir el tiempo de búsqueda, podemos trabajar con arrays o listas ordenadas. De este modo los tiempos serás:

* El tiempo de búsqueda se reducirá a O(n\*log n)
* El tiempo de la inserción aumentara a O(n), ya que tendremos que insertar el nuevo elemento en la posición indicada para que el array siga ordenado.
* El tiempo de eliminación seguirán siendo el mismo (O(n)).

Si queremos reducir los tres tiempos para tener una manera mas eficaz, tendremos que utilizar un árbol ordenado binario (BST), que los tiempo de búsqueda, inserción y eliminación van a ser de orden O(n\*log n).

Un árbol binario ordenado se caracteriza por tener como subnodo izquierdo un elemento mas pequeño, y como subnodo derecho uno mayor.

Para realizar las búsquedas, vamos a comparar con la raíz del árbol, para saber si es mayor o menor, y buscar en la parte correspondiente. Haremos esto sucesivamente con cada subnodo hasta llegar al elemento que queremos buscar.

Para insertar elementos, vamos a comparar el elemento con la raíz, y dependiendo de si es mayor o menos, seguiremos recorriendo el árbol por la rama correspondiente. Hasta llegar al ultimo nodo, donde lo insertaremos a la izquierda o derecha, según nos indique la comparación.

Para hacer la eliminación de un nodo, tenemos varias posibilidades:

* Sustituirlo por un nodo hoja, es decir, uno de sus subnodos que no tenga hijos
* Sustituirlo por el valor mínimo del subárbol derecho
* Sustituirlo por el valor máximo del subárbol izquierdo.