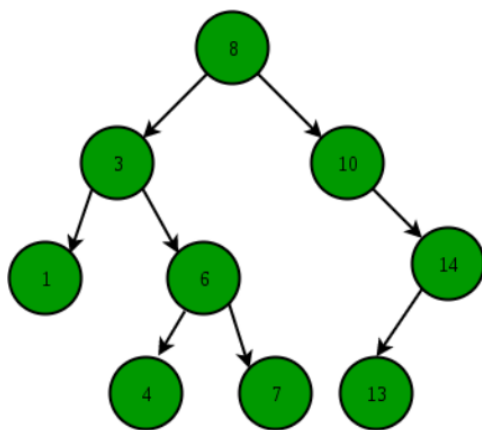


### Algoritmo de ordenamiento:

#### Binary Search Tree:

Es un árbol binario basado en nodos es una estructura de datos en la cual el valor del nodo de la izquierda es menor al valor del padre o raíz. El de la derecha es mayor al del padre. Estos nodos [izquierda y derecha] también deben ser un árbol binario de búsqueda.

La complejidad de este algoritmo se ve determinada por su altura, y es igual a  $O(\log n)$ . Es mucho mejor trabajar con un árbol binario de búsqueda balanceado, esto significa que cada subárbol está completo. Por ejemplo, la imagen de abajo no contiene un árbol binario balanceado.



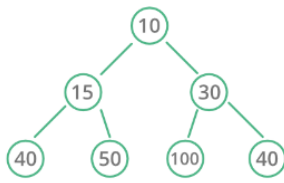
#### Heap + Heap Sort:

Heap sort es similar a selection sort, se busca el elemento más chiquito y ponerlo en el inicio, a partir de allí se empieza el sorteo. Es simple, usa poca memoria, y es eficiente. Un heap es una estructura de datos de árbol.

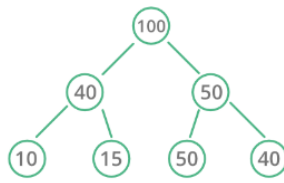
El algoritmo heap tiene diferentes funciones que cambian de complejidad dependiendo de dónde se están ejecutando. El peor caso de un binary heap, es **insert =  $O(\log n)$** , **delete-min =  $O(\log n)$**

Para resolver la SP me apoyé bastante del algoritmo heap, y su implementación.

## Heap Data Structure



Min Heap



Max Heap