

Dada la siguiente secuencia de números: 8, 9, 11, 15, 19, 20, 21, 7, 3, 2, 1, 5, 6, 4, 13, 14, 10, 12, 17, 16, 18, generar:

a) Su árbol binario de búsqueda:

Sabemos que un árbol binario de búsqueda es un árbol binario ordenado, así que lo primero que haremos será ordenar los datos.

$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21\}$

Sabemos que un árbol binario es aquel en el que cada nodo tiene como máximo grado 2.

También sabemos, que para que nuestro coste computacional sea menor, nuestro árbol tiene que tener la menor altura posible.

Para conseguir esto, vamos a contabilizar los datos y encontrar la mediana, pero como varicara el resultado dependiendo de si el número de datos es par o impar lo haremos así:

- Si es impar: $\frac{\text{Numero total de datos}}{2}$, nos quedamos con la parte entera y le sumamos 1.

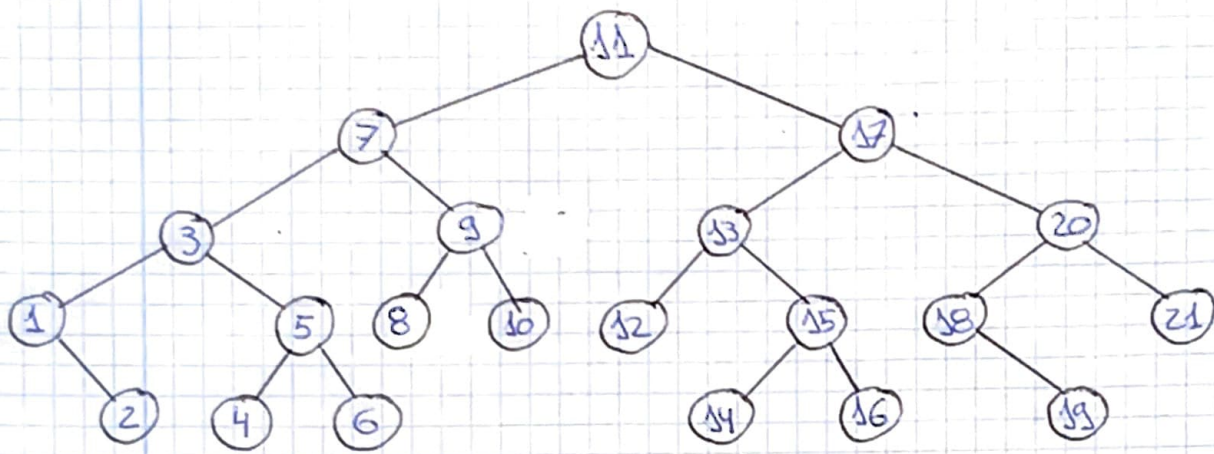
- Si es par: $\left(\frac{\text{Numero total de datos}}{2}\right) + 1$.

En nuestro caso tenemos 21 datos, que son impares:

$$\frac{21}{2} = 10,5 \text{ cogemos la parte entera y sumamos 1.}$$

$10 + 1 = 11$ este será nuestro nodo raíz, que en este caso coincide con el número 11 porque es una serie numérica de 1 a 21 ordenada.

La ordenación a utilizar es inorden, así que, como nuestro nodo raíz es el 11, el subárbol izquierdo contendrá los datos del [1 al 10] y el subárbol derecho del [12 al 21].

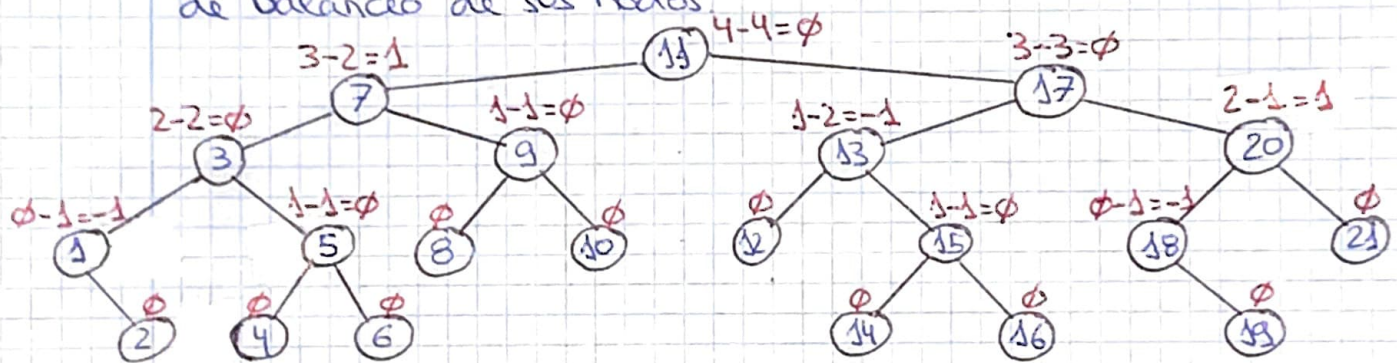


Este sería nuestro árbol binario de búsqueda.

b) Si árbol perfectamente balanceado:

Un árbol balanceado, es un árbol en el que la diferencia en altura entre el subárbol izquierdo y derecho es 1, 0 o -1.

Vamos a analizar nuestro árbol analizando el factor de balanceo de sus nodos.



-Todas las hojas tienen coste 0.

-Para el resto de nodos: altura subárbol izquierdo menos altura subárbol derecho.

Podemos decir que nuestro árbol está perfectamente balanceado.

c) Si árbol AVL:

Un árbol AVL es un árbol binario ordenado y balanceado.

En nuestro caso, nuestro árbol es un árbol AVL.

¿Qué conjeturas puedes sacar de estos ejercicios?

La principal conjetura que puedo sacar, es que antes de ponernos a trabajar con un árbol, debemos ordenar los datos.

Al ser árboles binarios, el nodo raíz casi siempre rondará en el valor central de los datos en el recorrido en orden.

¿En que tipo de árbol se genera el árbol de altura mínima?
En los árboles balanceados, ya que para que este balanceado, la diferencia entre la altura de los nodos del subárbol izquierdo y el subárbol derecho debe ser $1, 0$ o -1 .

¿Cuáles serían las ventajas y desventajas de uno y otro esquema de representación de árbol?

Un árbol de búsqueda tiene la ventaja de que sus elementos están ordenados, pero como no necesariamente tiene que estar balanceado, su coste computacional será mayor que el de un árbol balanceado, el cual, tiene la ventaja de ser el tipo de árbol que genera menos altura y por lo tanto mayor eficiencia.

Pero los árboles AVL, son árboles balanceados y ordenados, con lo que cogen las ventajas de los dos árboles anteriores. Mejor eficiencia por la menor altura del árbol y mejor eficiencia en la búsqueda de datos al encontrarse ordenado. Al estar balanceado y ordenado, cuando realicemos una búsqueda, podemos ir descartando la mitad de los datos en cada paso de nodo.

Sabiendo poco de programación, supongo que una desventaja será que cada vez que insertamos o eliminemos un dato, su coste crezca al tener que reordenarse y balancearse.