Binary Search Tree

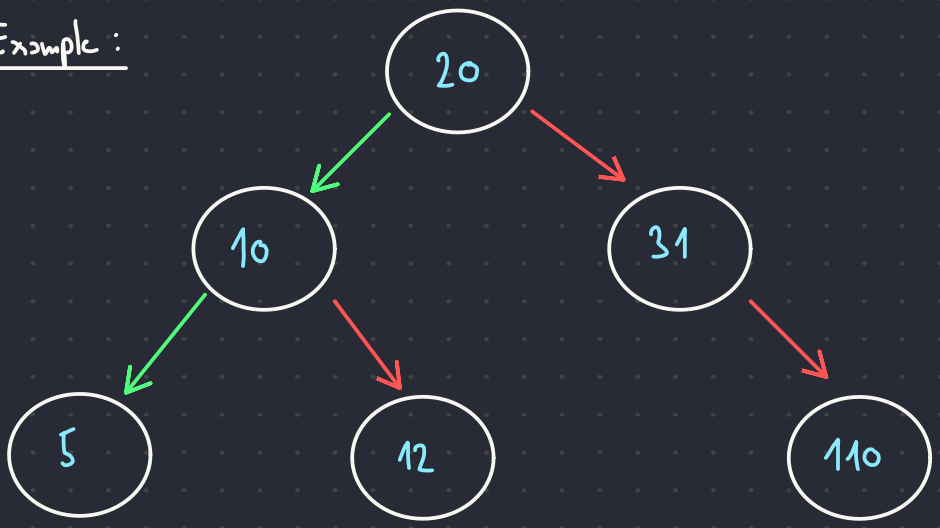
1) All nodes have at most 2 children -> Left and Right child

2) All keys in left subtree are less than the key in current node
   and all keys in the right subtree are greater

3) No duplicates are allowed

- Left = Lesser
- Right = Greater

Example :



How does * and & operators work in C ?

& (Address-of Operator)
- gives address of variable in the memory

* (Dereference Operator)
- gives value stored at address

For example:

int x = 10; (variable with value 10 of type int)

int * pt_x = &x; (variable with value like 0x7ffc35f5f80c of type int *)

*int (gives us 10)

Our usecase:

We have local variable Node * root; (example value 0x5cfc33d5f98b)

If function takes Foo(Node * root) new variable will be created and
the value (0x5cfc...) will be copied. So if we change the copy, nothing happens
to the original one.

But image we have Node * root = NULL.
After the first Insert, we would like the root not to be NULL.

So we need Insert to take Node ** root. Thats the memory location of the
Node * variable, so it can access the meory where this variable is stored and
assign new value

*root = (Node *) malloc(...);