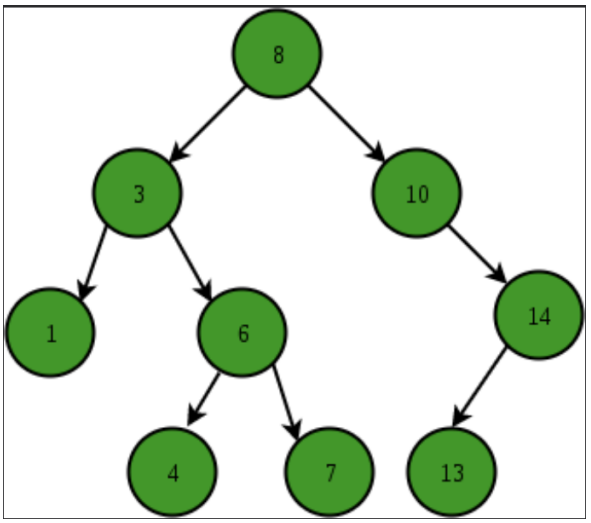
Tasks from “Algorithms”

3.2.1: Insert in binary search tree



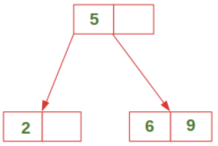
When inserting a new node into a binary search tree you always have to check whether the new node's value is bigger or smaller then the node you are currently on.
Let's say you wanted to insert 9 into the binary search tree on the right. First you would ask - “is 9 bigger or smaller than 8?”. Depending on the answer you either move down the left leg or the right, in our case it's bigger so we go down to the right. We then ask again if the value we want to insert is bigger or smaller then the new node we stand on. In our case its smaller so we move down to the left, but given there is no node there we instead insert our new node.

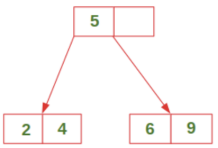
3.2.18: Delete in binary search tree

When deleting a node in a binary search tree there are a few things to keep in mind. In case the node you want to delete is a leaf node (it doesn’t have any nodes below it), you can simply delete it from the tree. However if the node is a parent node, you have to relocate the child nodes so the values are placed correctly. If we wanted to remove 3 from the image above we would move 6 up and replace 3, but we would also have to relocate 1. In this case 1 would be moved down and have 4 as a parent node instead.

3.3.1 : Insert in 2-3 tree

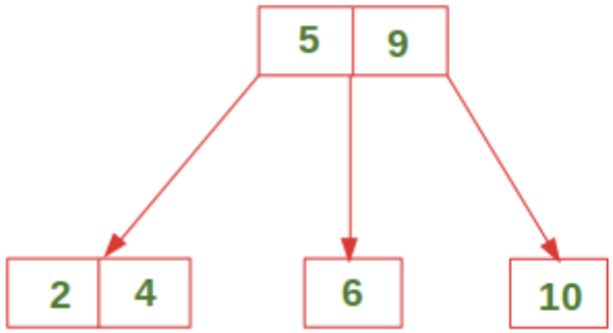
When we insert a new value into a 2-3 tree there are 3 cases you need to look out for.
The first case is when we want to insert a value into a tree with only one data element. In the example on the right we want to insert 4 into the tree and given there is only one element we simply add the value on the left side of 5 as 4 is less than 5 and on the right side of 2 as 4 is greater than 2.



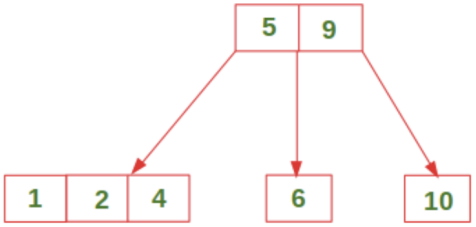


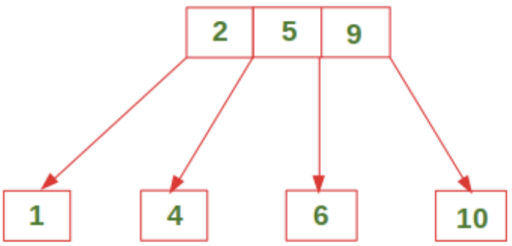
The second case is when the data element contains two values but the parent only contains one. Let's say we wanted to add 10 to the tree above, it would go on the right side of 9. Now the right leg would contain 6, 9, 10 but the parent element would

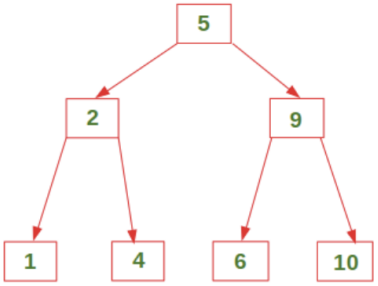
still only be 5 so to balance the tree you relocate the middle value up to become a parent element like 5. This means the new tree would look like this.



The third case is when you want to insert a new value into a tree where the node and the parent both have two elements. Again working from the tree above but now we want to insert 1 into the tree. 1 is less than 5 and less than 2 so it goes into the bottom left node on the far left side. Now the node contains 3 values so we move the middle value 1 layer up but that also makes the parent node have 3 values so we now mode the middle value from the parent node 1 layer up. This can be seen in the images below.







3.3.10 :Insert in red-black binary search tree

When inserting into a red-black tree there are two things you need to think about. The first thing is recoloring and the second is rotation. As the name says nodes can either be red or black and when you insert a new node it's always red.

Recoloring means when you insert a new red node and the parent node is also red, you have to recolor the parent node to black and afterwards do the same check all the way up the tree.
Rotating is about keeping the balance of the tree. This means every time you add a new node to the tree you do a rotation to keep both sides of the tree “evenly weighted”.