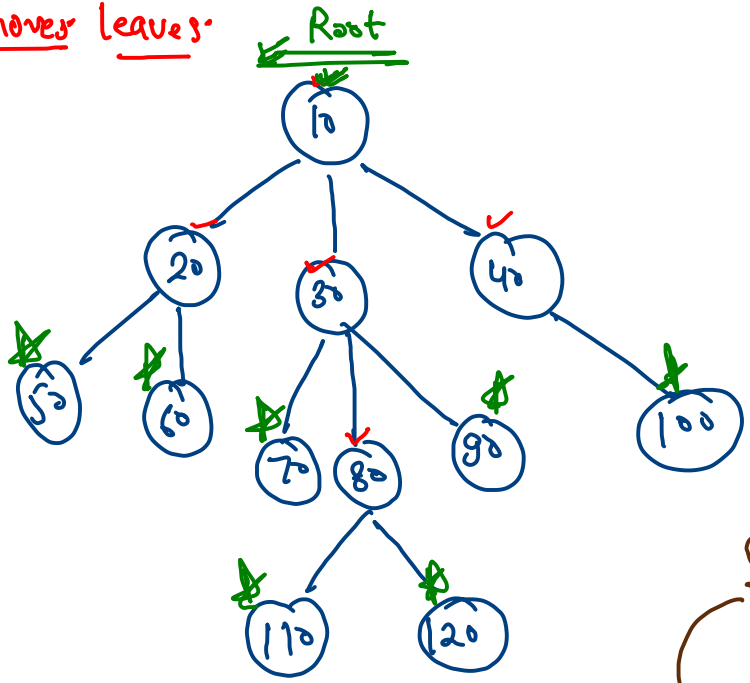
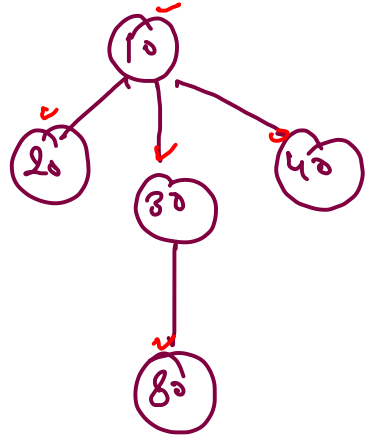
Remove leaves.



After removal of leaf



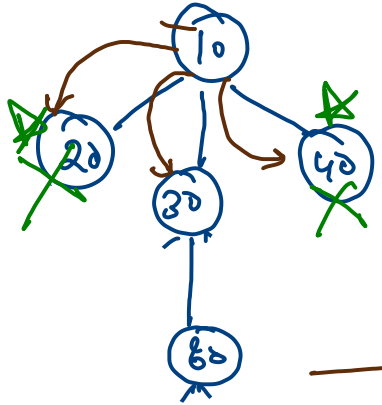
Expectation → Root → removeleaf

Faith → child → remove your leaf } Gt will work.

→ How to merge??

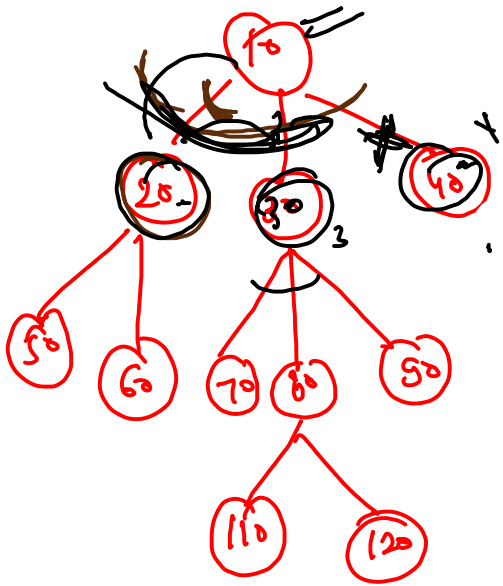
leaves → Node having 0 child is know as leaf

node.child.children.size() == 0 } node.child is leaf.



→ Children.

Pre Area. → } at dom change
Post Area → } the flow of code.



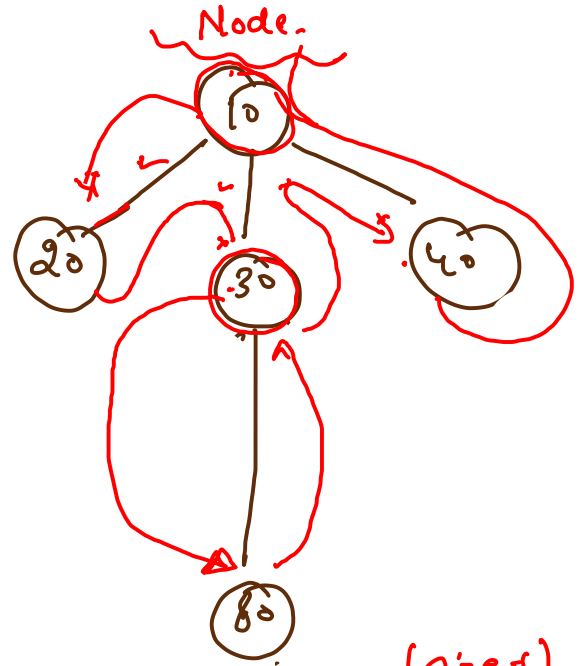
You are leaf on my level.

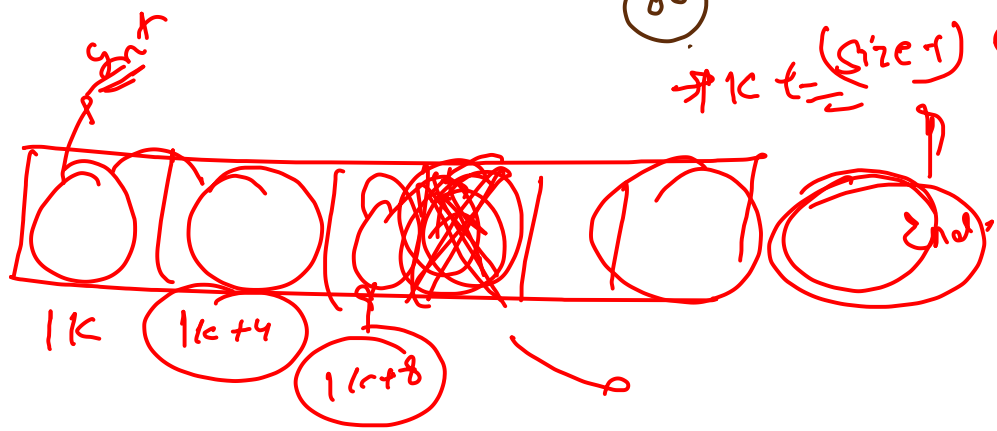
Post Area
- faith → (child) → to remove leaf
- Remove leaf from our level.

Pre Area
- Remove leaf from our level.
- faith → (child) to remove leaf

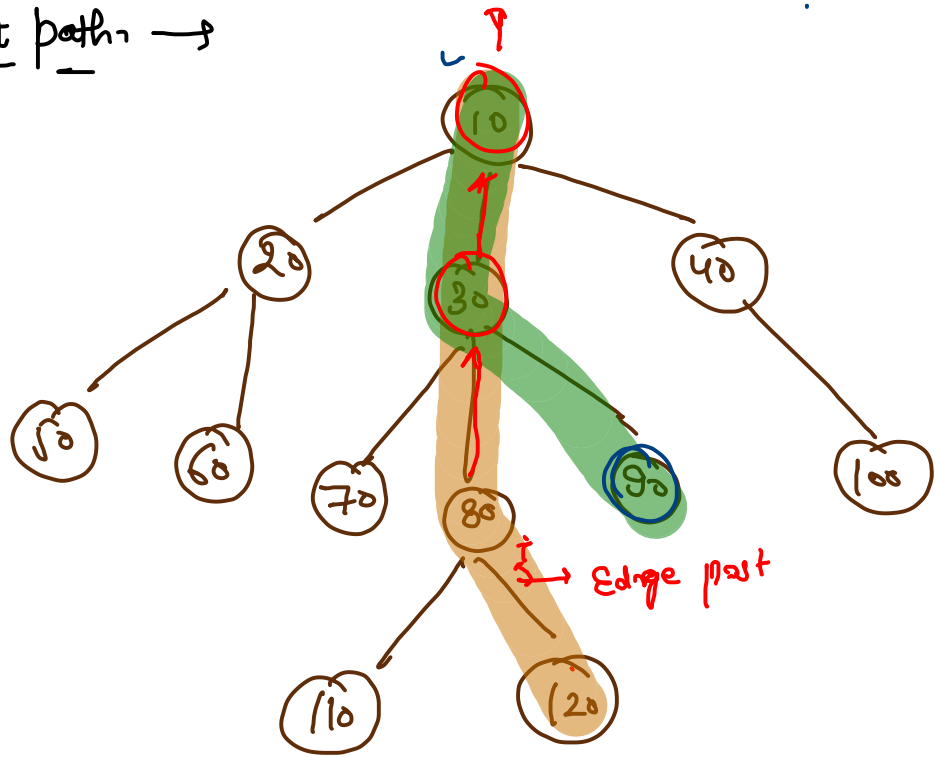
```java
for(int i = 0; i < root.children.size(); i++) {
    Node child = root.children.get(i);

    if(child.children.size() == 0) {
        root.children.remove(i);
        i--;
    }
}

// call
for(Node child : root.children) {
    removeLeaves(child);
}
```

Modification





Node to Root path →



120

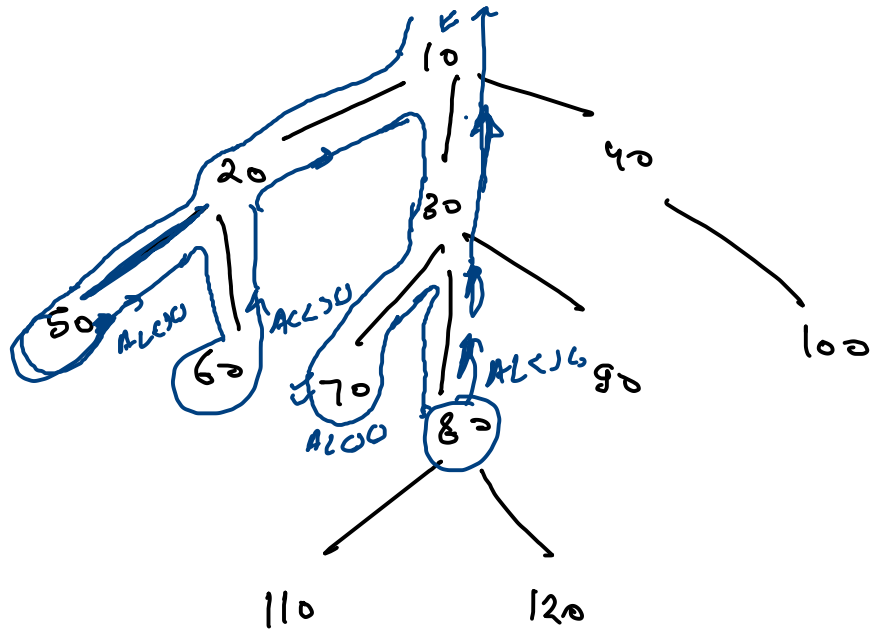
120

120 → 80 → 30 → 10

90 → 30 → 10

80 → 30 → 10

return type.



80

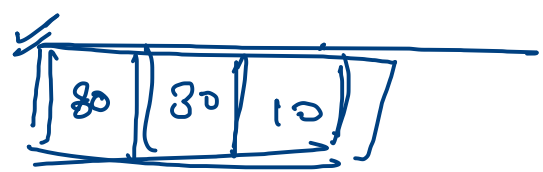
node2rootpath

→ AL<>()

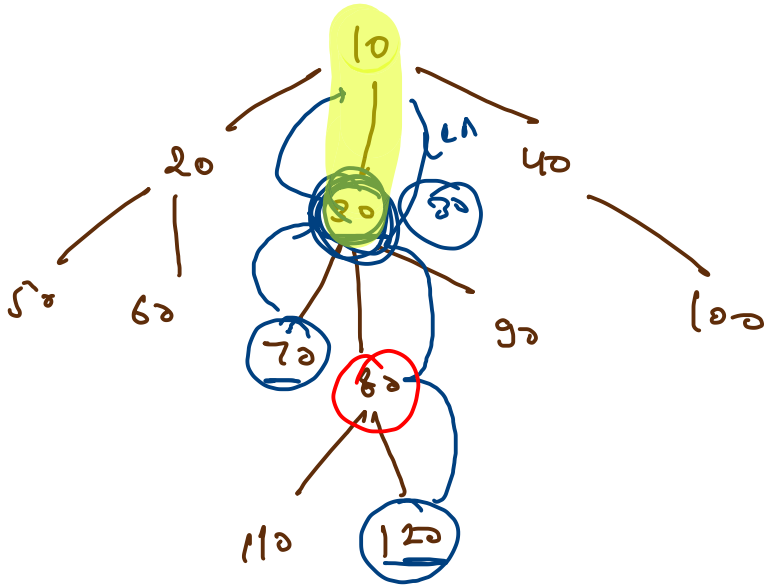
size 20

```
if( node.data == data) {
    AL<Int> bres = new AL<>();
    bres.add(80);
    return bres;
}
```

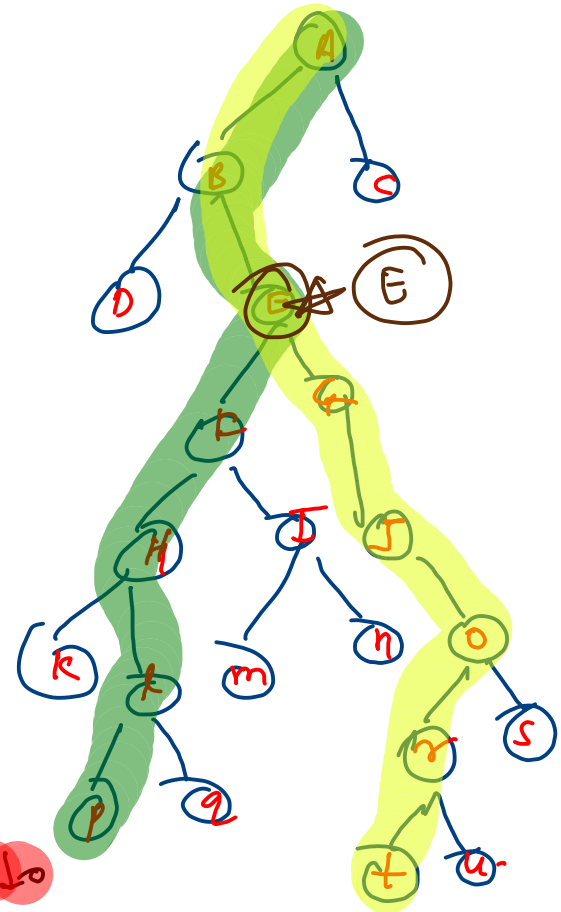
```
rres = node2rootpath(child);
if(rres.size() == 3) {

}
```



Lowest common Ancestor } LCA

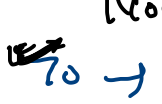


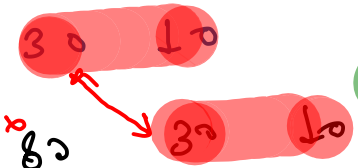
70
120 } LCA



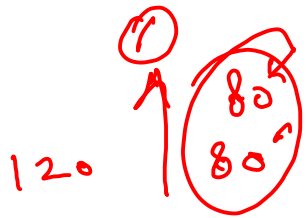
node to Root path

Node to root path





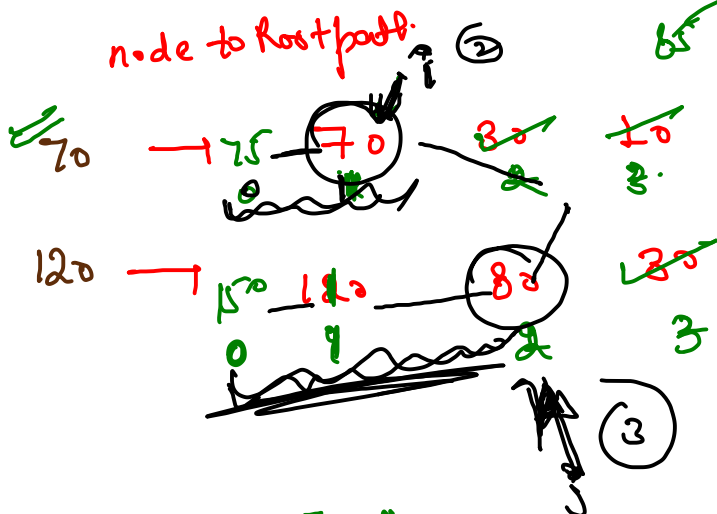
t
p } LCA



Distance between 2 nodes:

on the basis of Edge

node to Root path.



$i = 1$

$j = 2$

$i+1$

$j+1$

75 70 30

150 110 80 30