Supervision work
Supervision 14

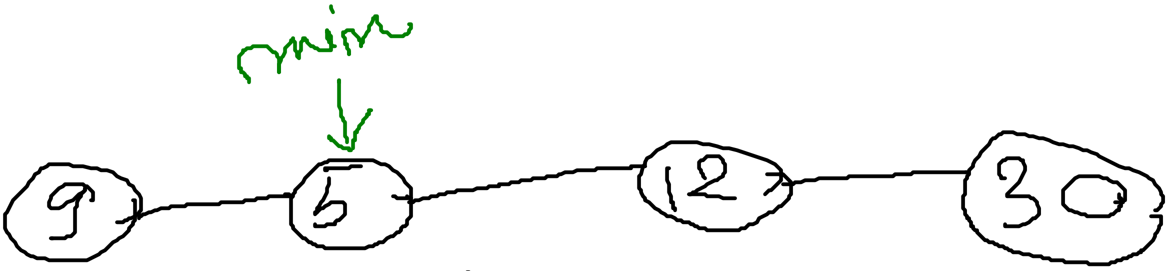
Tudor Avram
Homerton College, tma33@cam.ac.uk

23 Feb 2016

1 Fibonacci Heaps

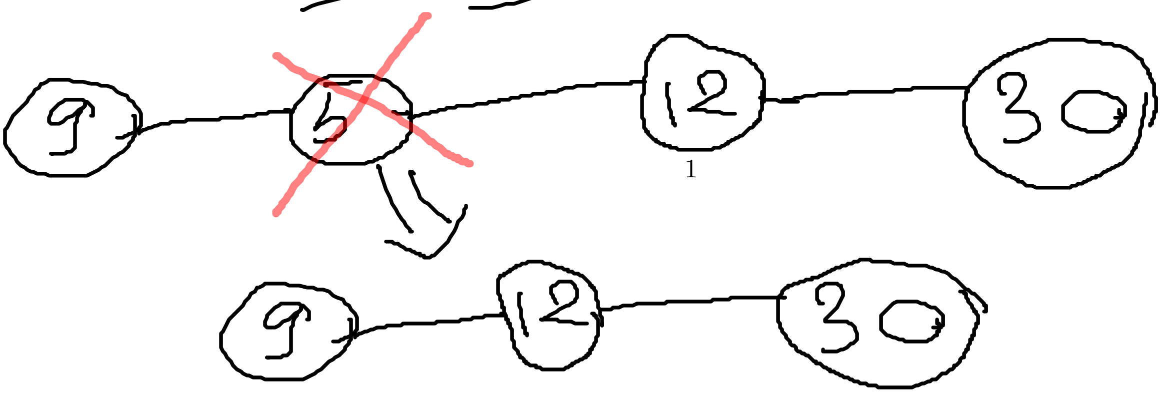
1.1 Exercise 1

(i) Insert 9, 5, 12, 30



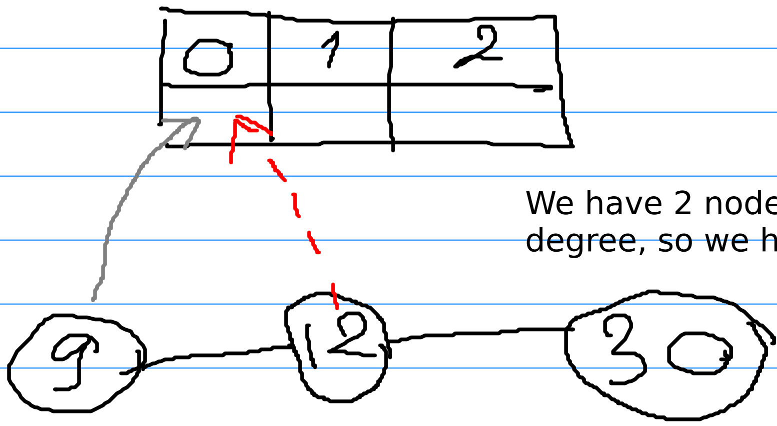
(ii) ExtractMin

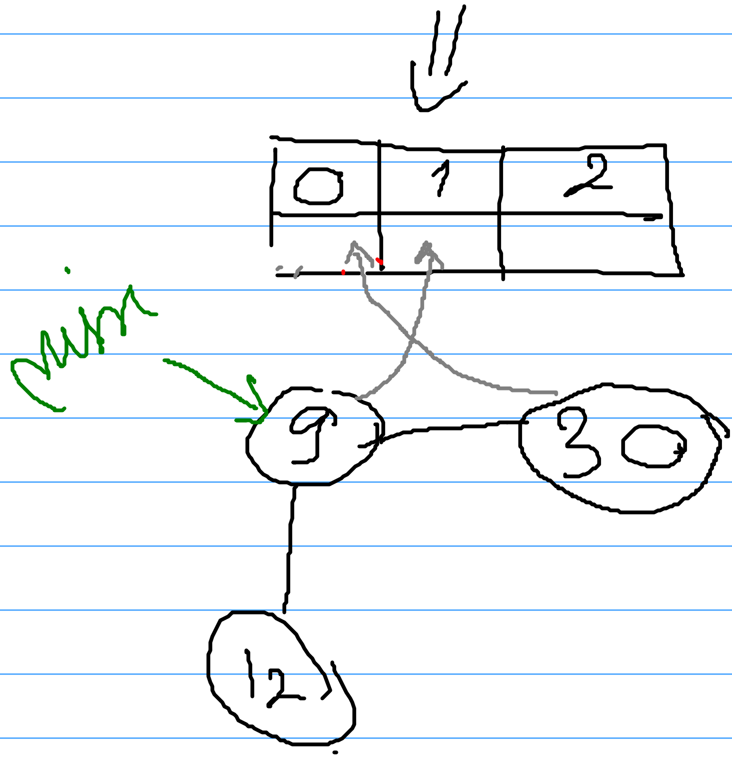
(a) Step 1: Extracting



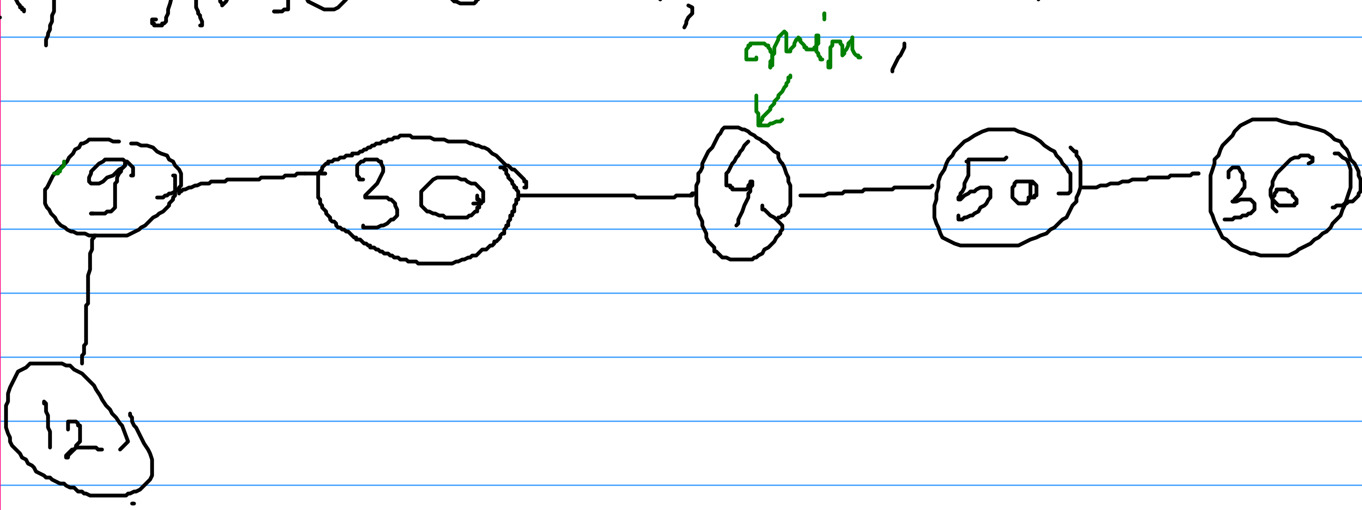
(b) Step 2 : Restructuring

We have 2 nodes with the same degree, so we have to merge them



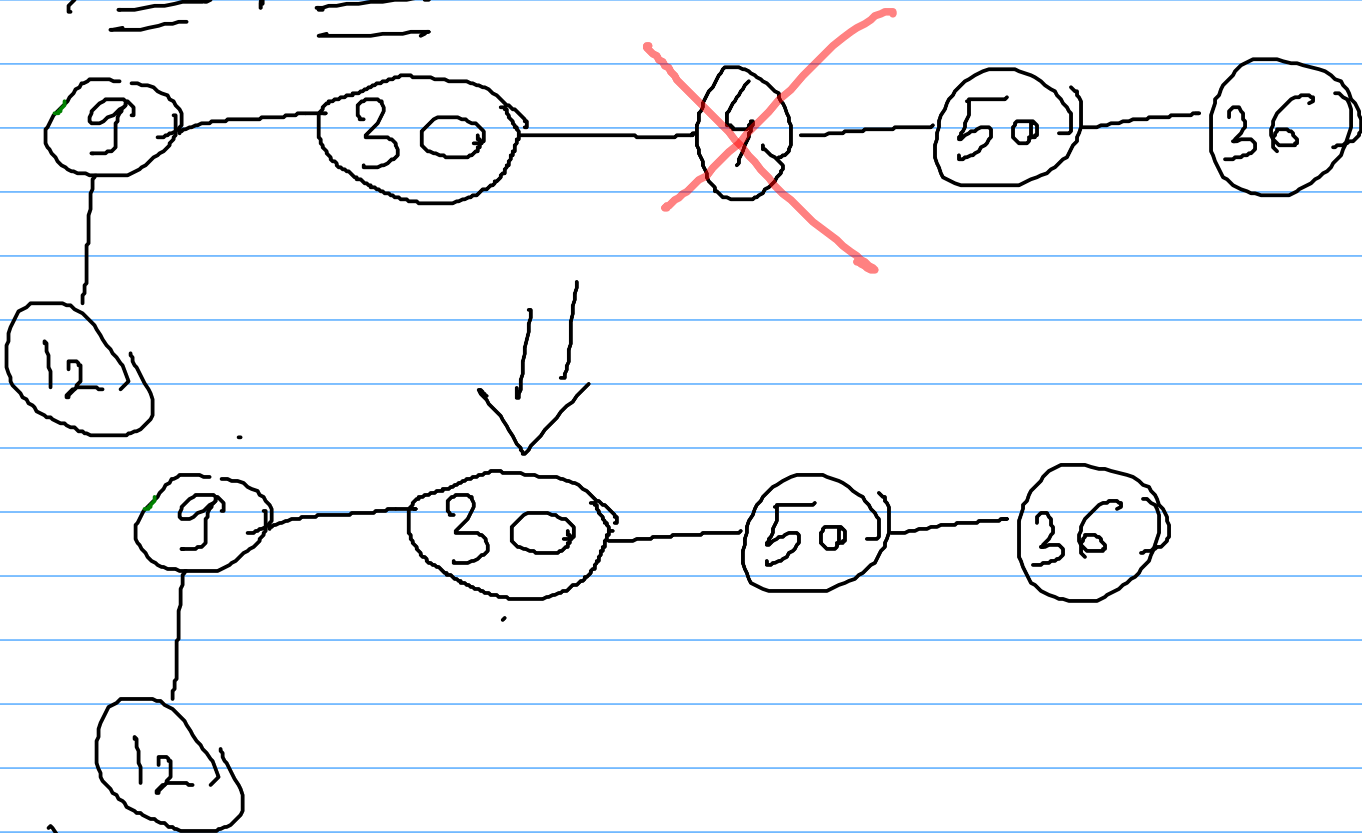


(iii) Insert 4, 50 36

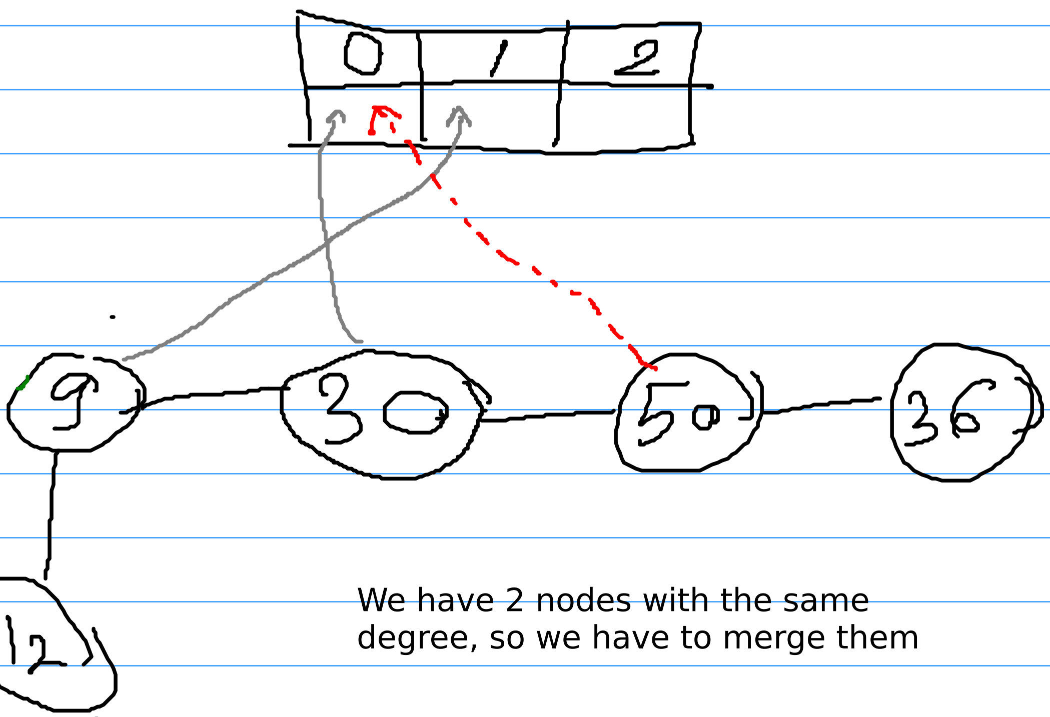


(iv) Extract Min

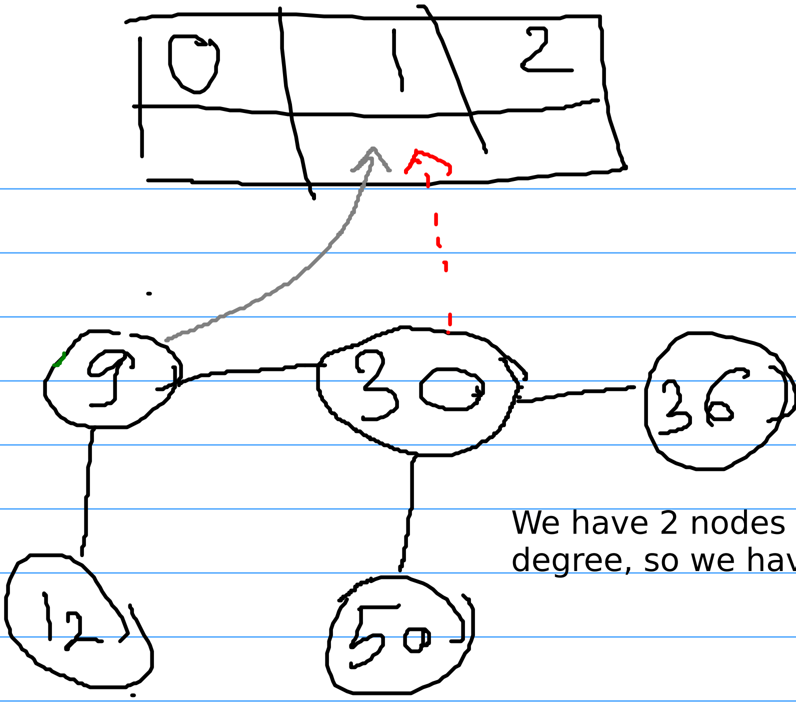
(a) Step 1



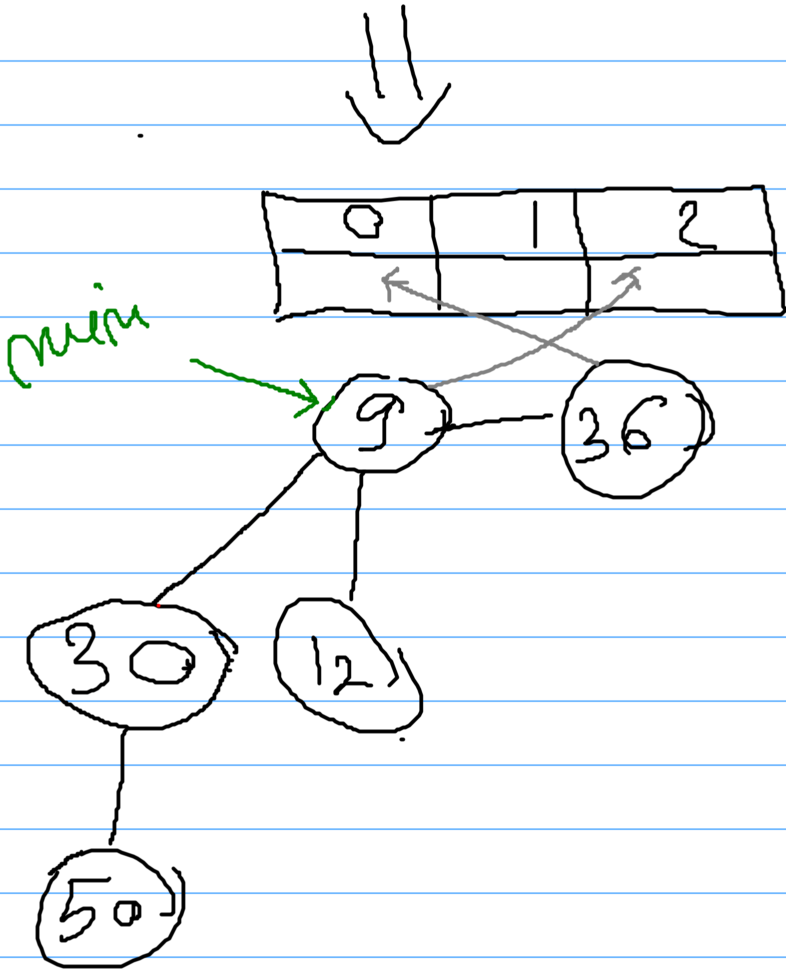
(b) Step 2



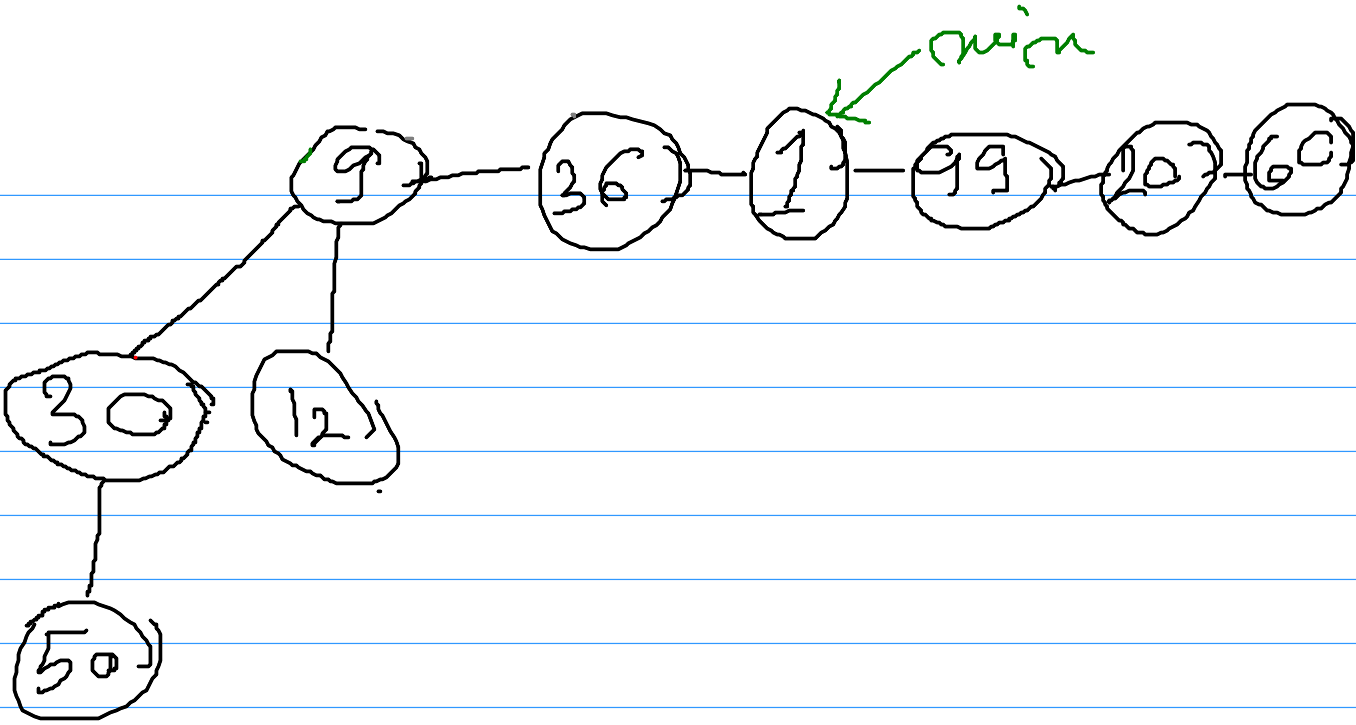
We have 2 nodes with the same degree, so we have to merge them



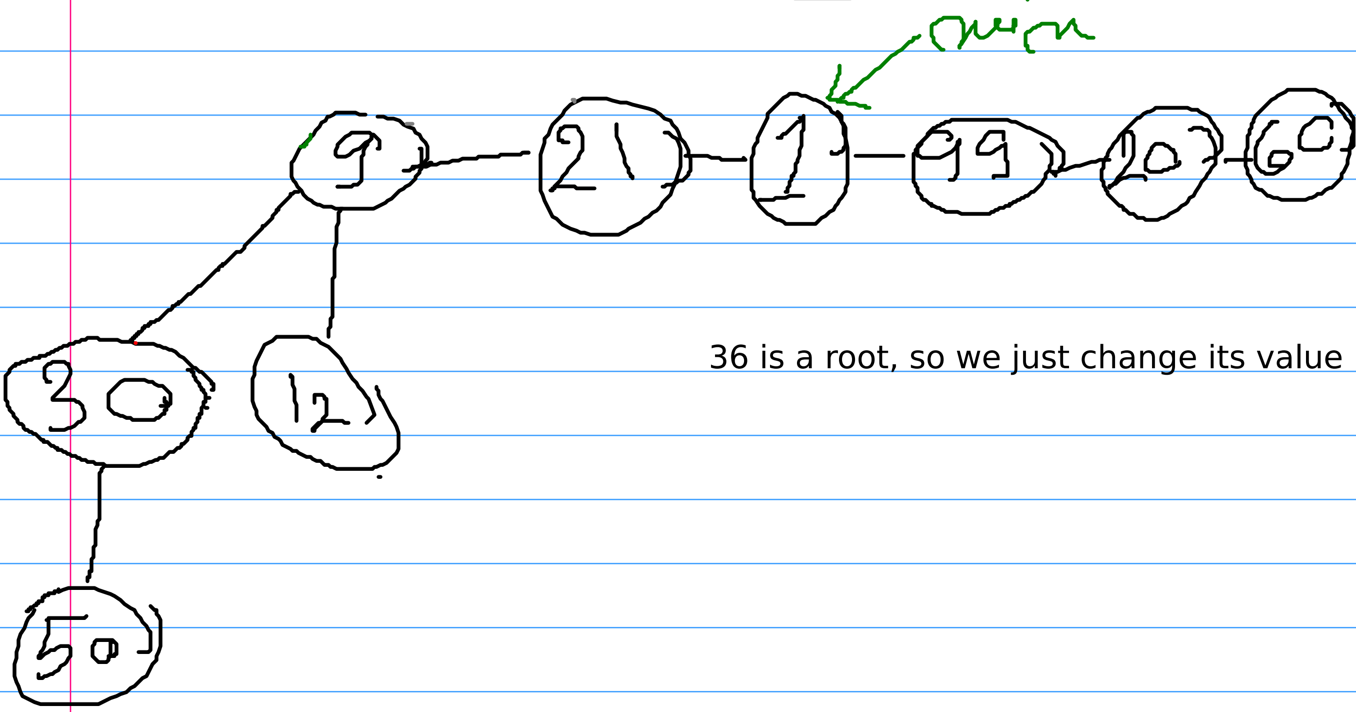
We have 2 nodes with the same degree, so we have to merge them



(v) Insert 1, 99, 20, 60

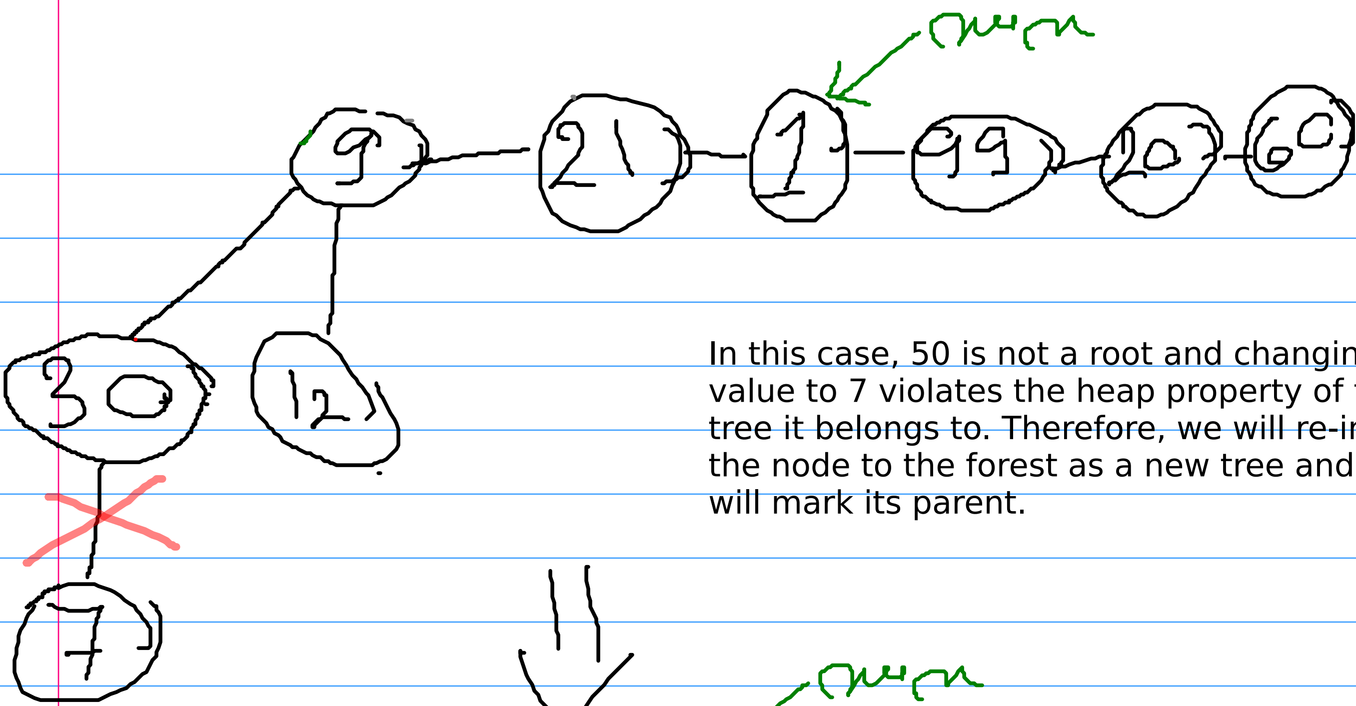


(vi) Decrease 36 → 21

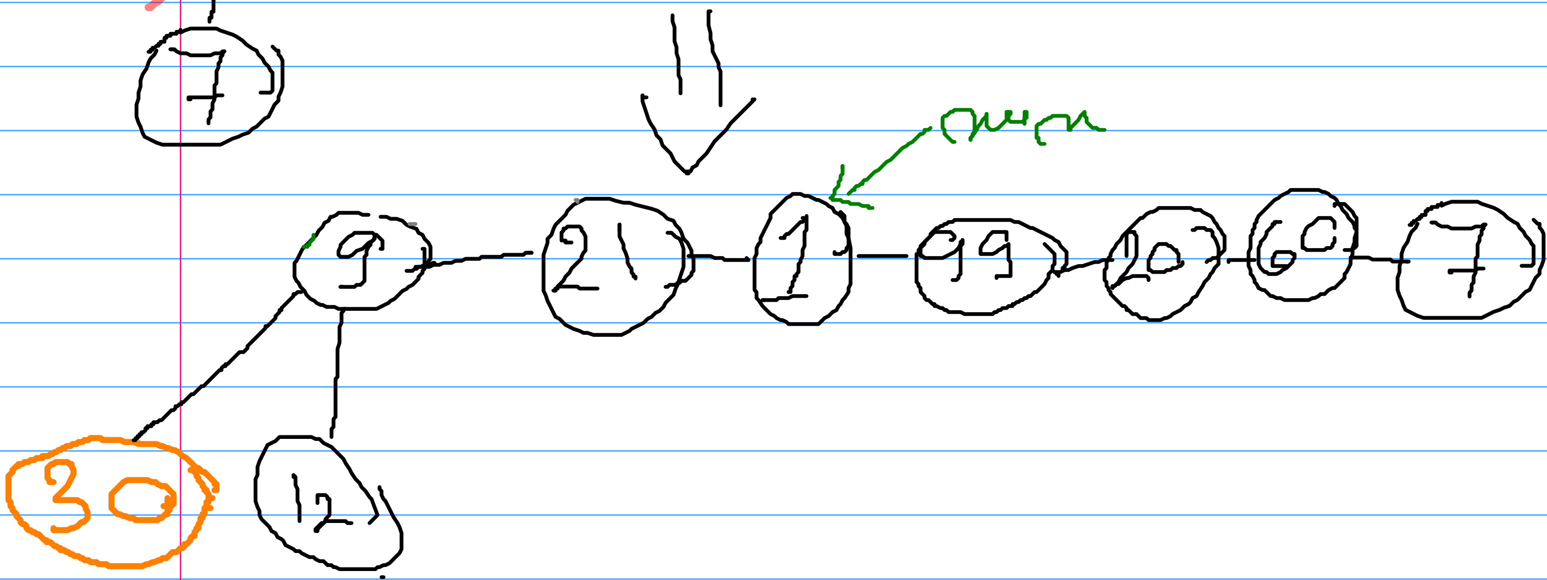


36 is a root, so we just change its value

(vii) Decrease 50 → 7

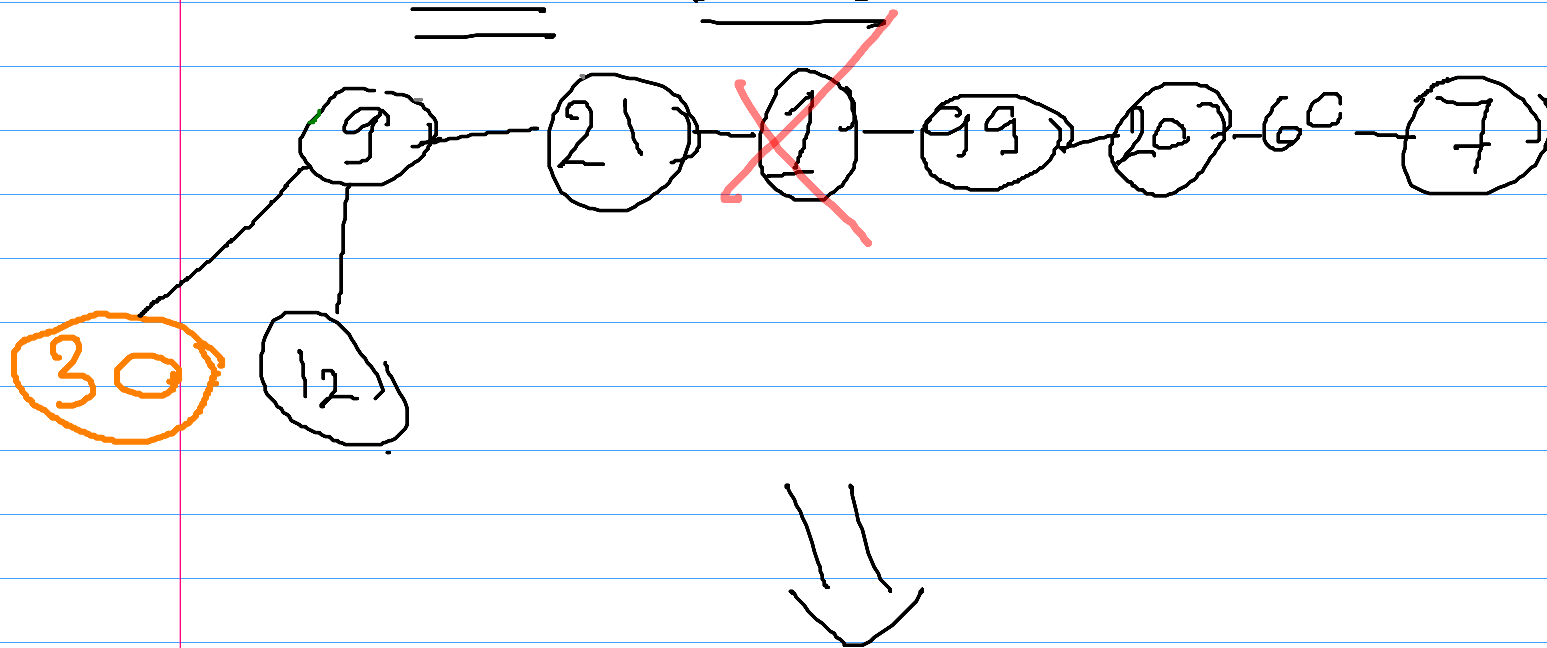


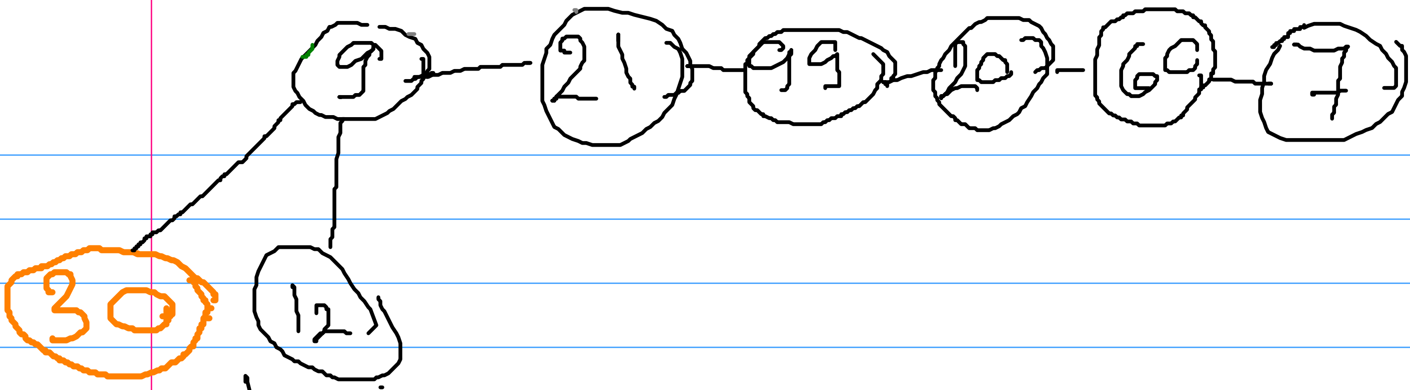
In this case, 50 is not a root and changing its value to 7 violates the heap property of the tree it belongs to. Therefore, we will re-insert the node to the forest as a new tree and we will mark its parent.



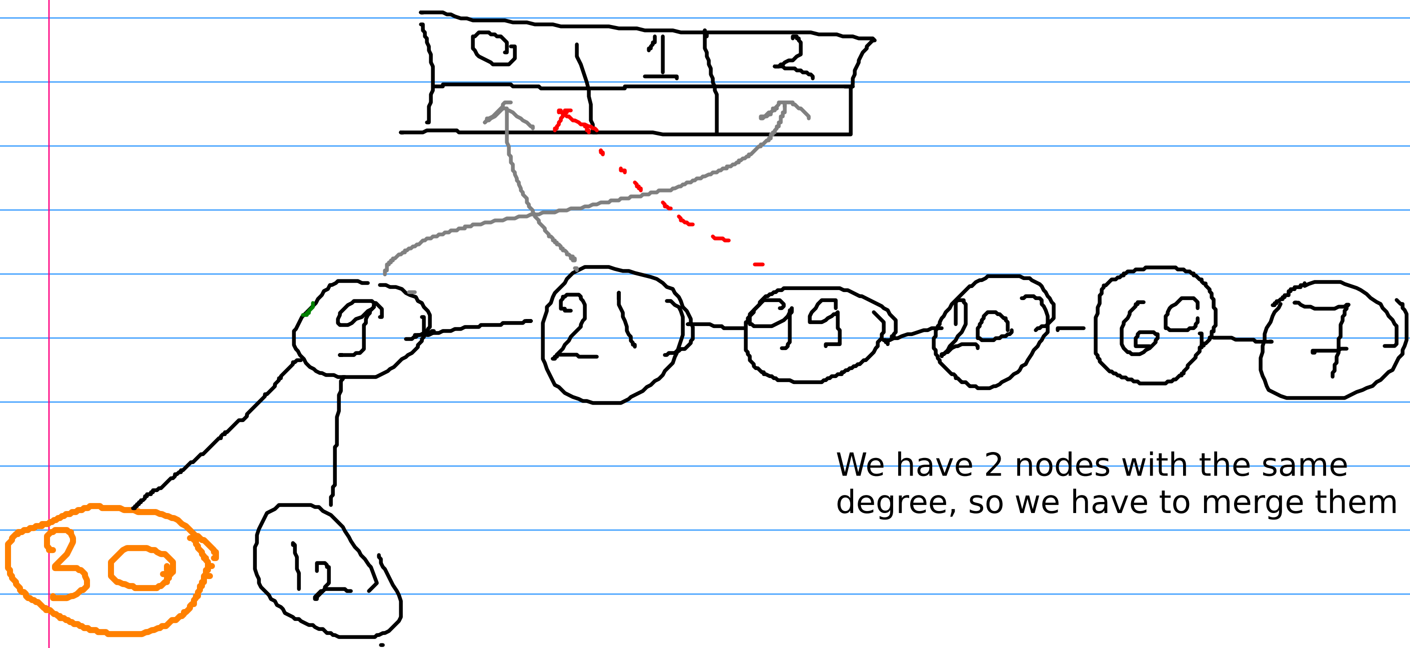
(vii) ExtractMin

(a) Step 1

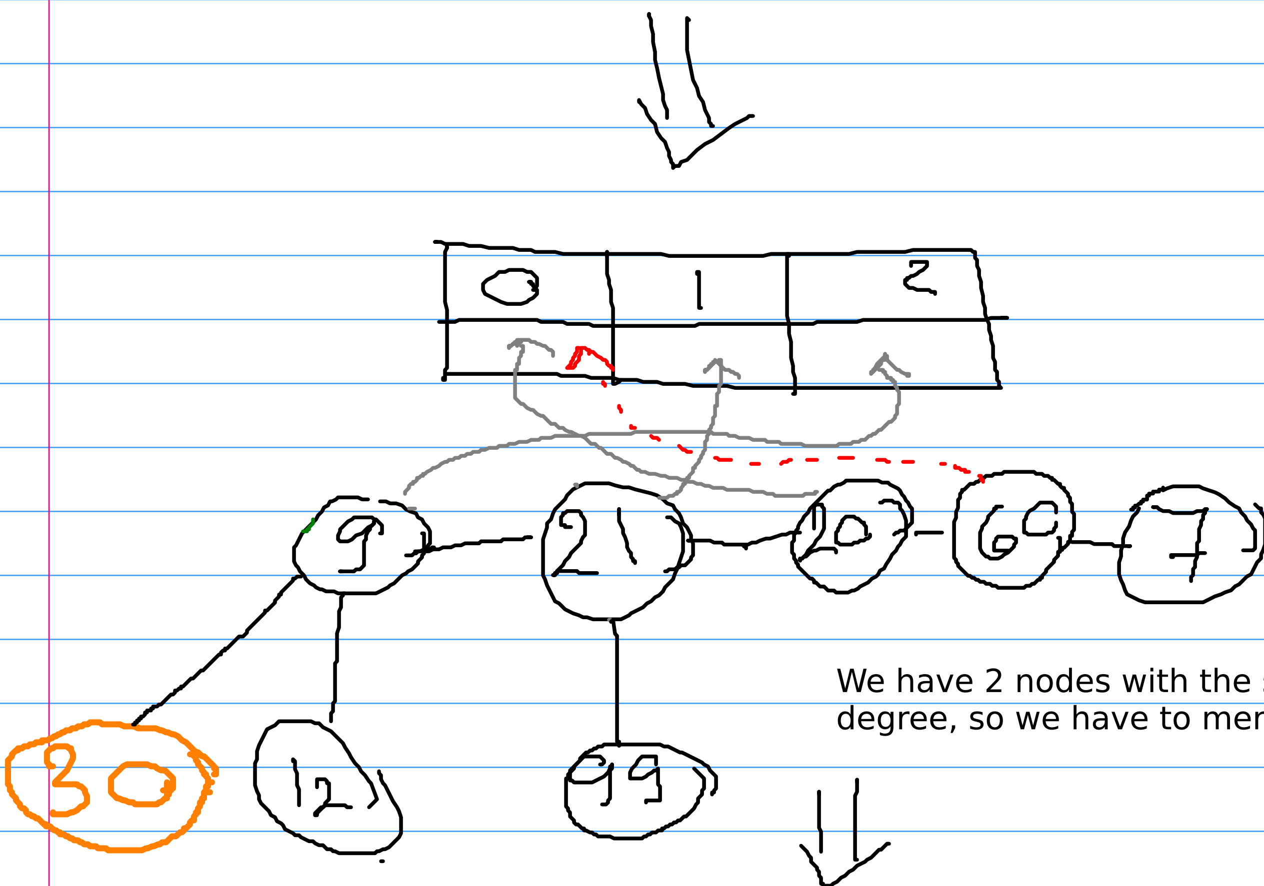




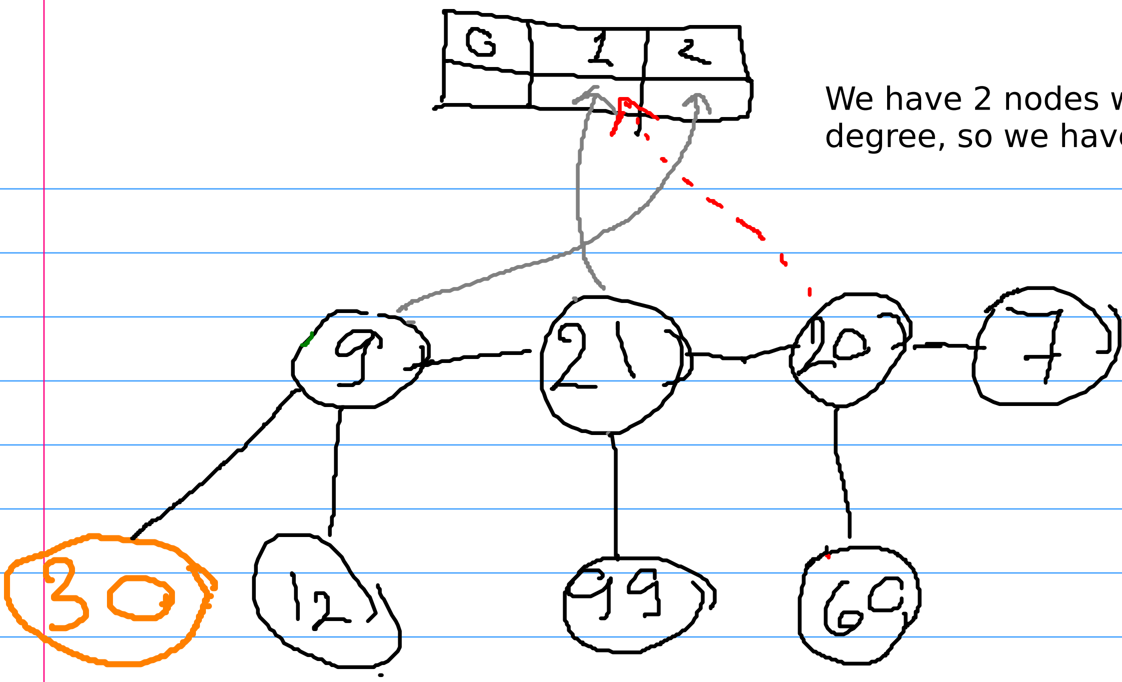
(b) Step 2



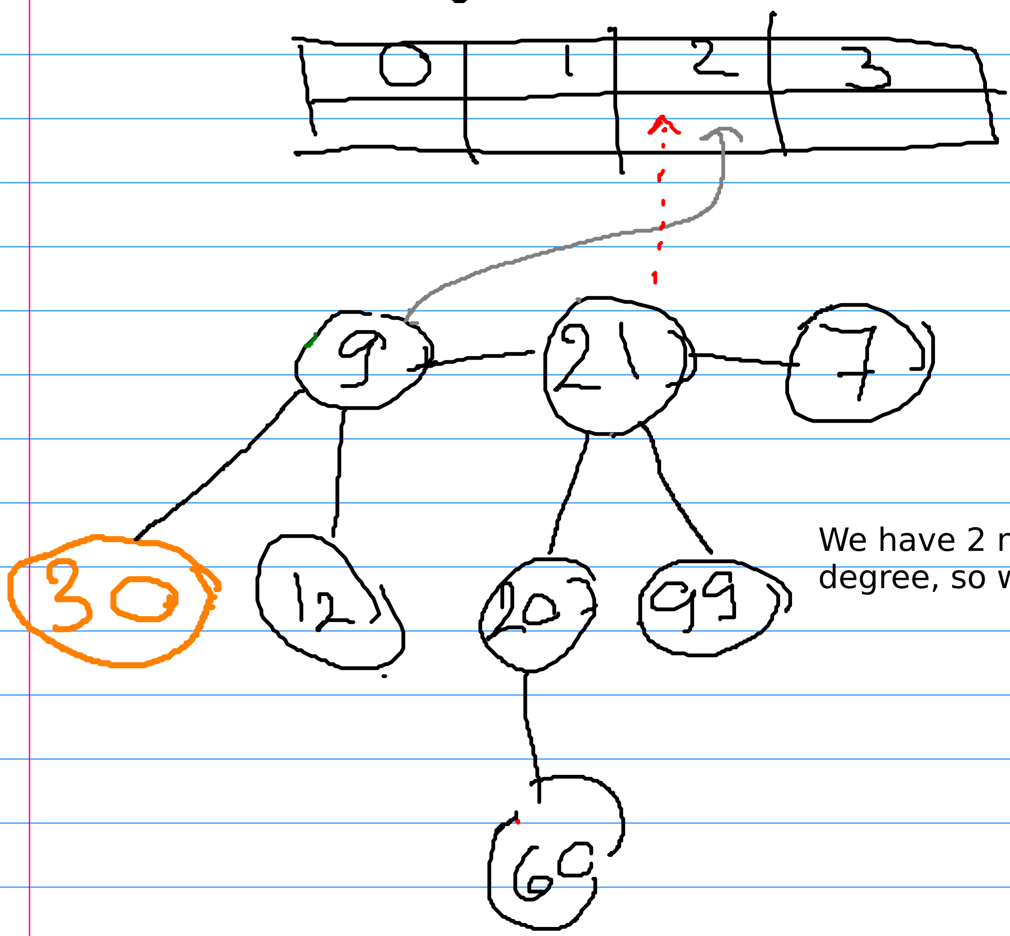
We have 2 nodes with the same degree, so we have to merge them



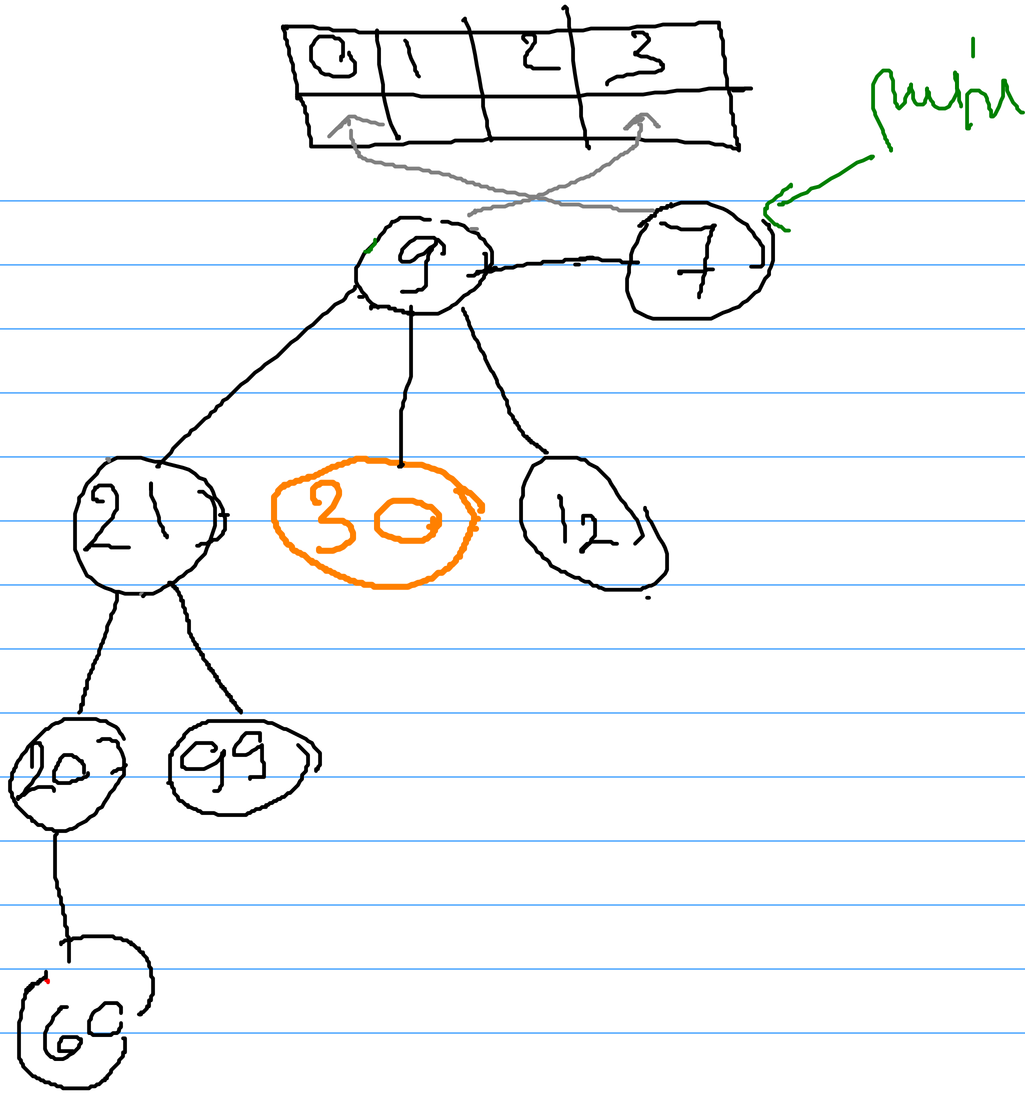
We have 2 nodes with the same degree, so we have to merge them



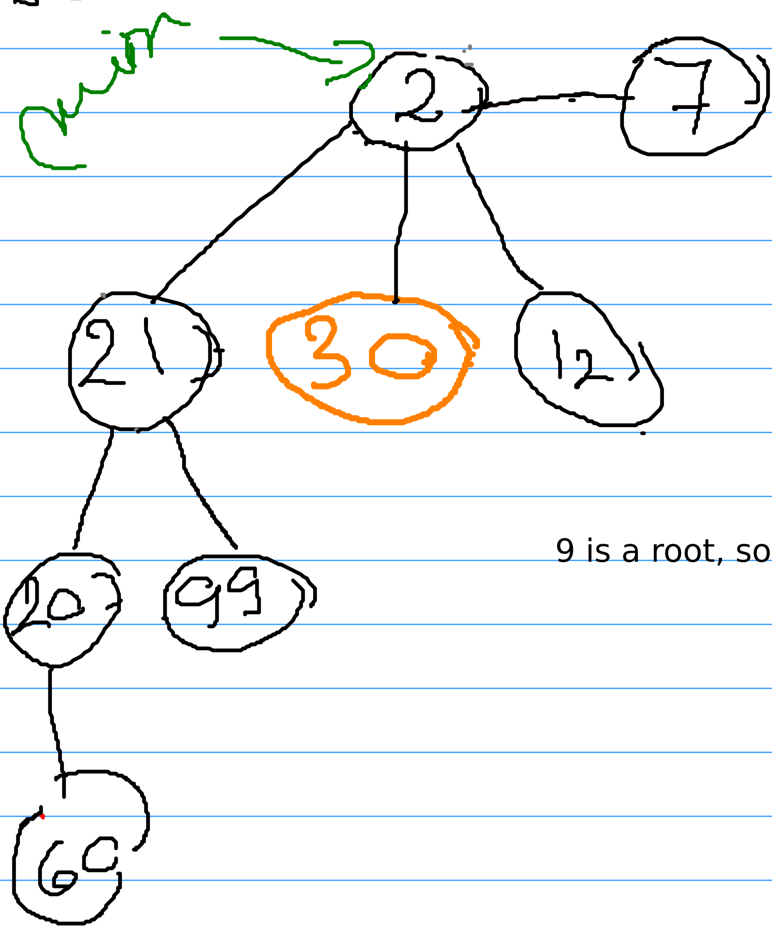
We have 2 nodes with the same degree, so we have to merge them



We have 2 nodes with the same degree, so we have to merge them

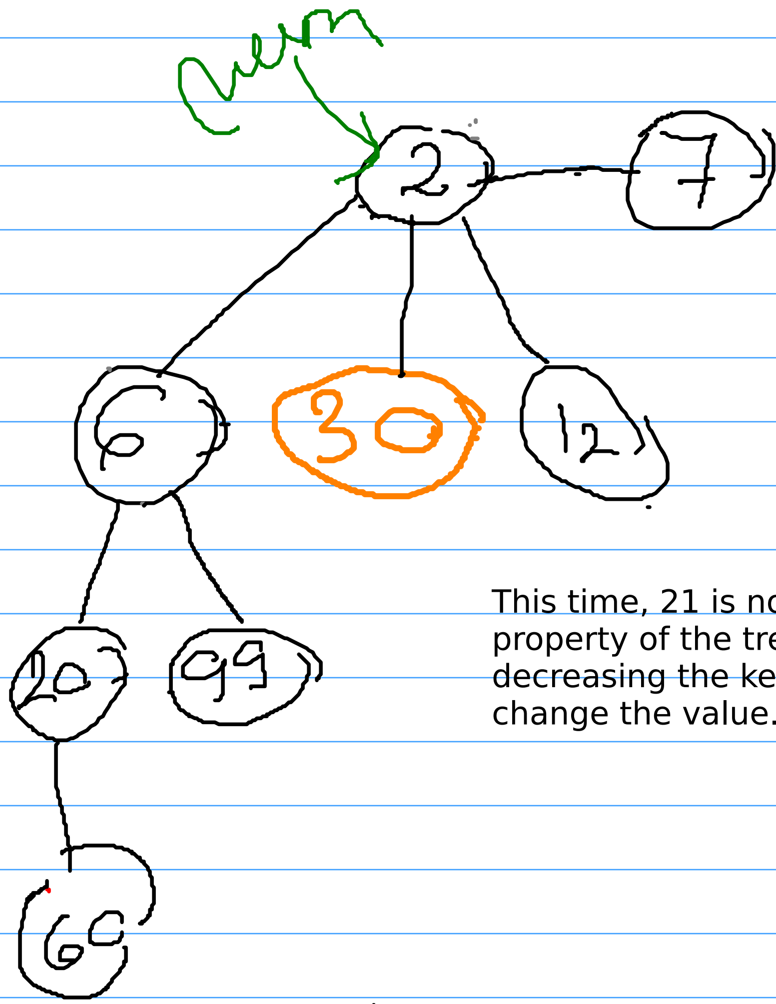


(ix) decrease 9 → 2



9 is a root, so there's nothing else to be done :D

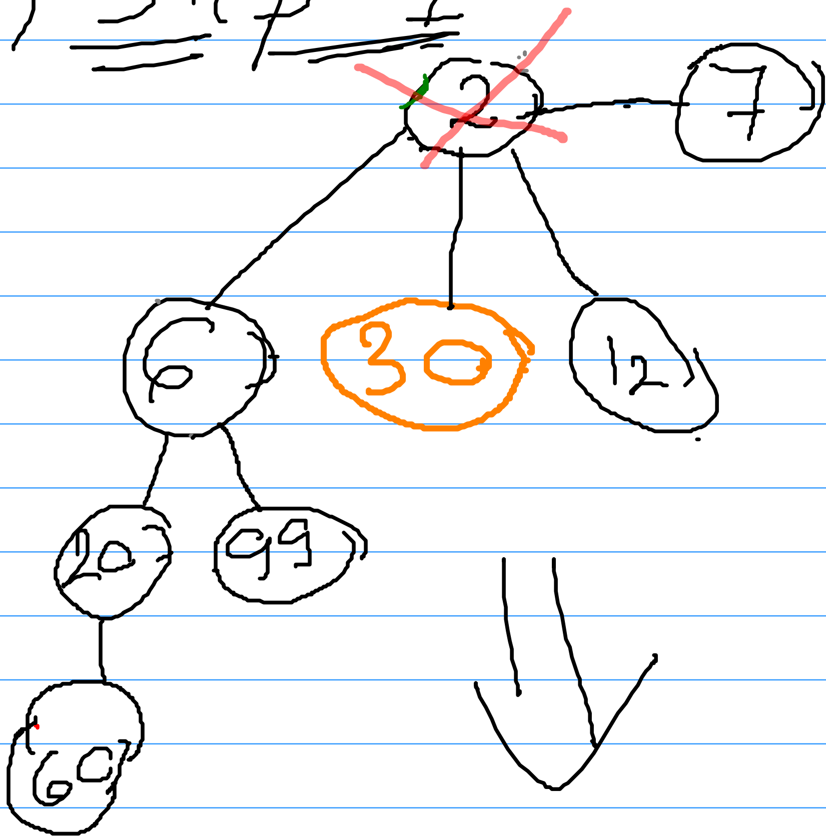
(x) Decrease $21 \to 6$

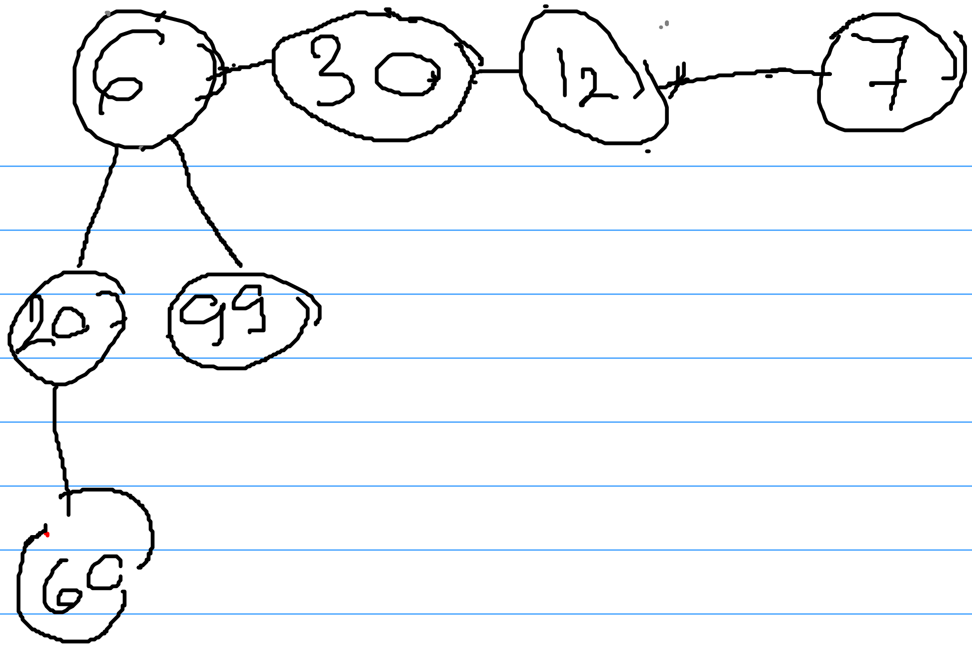


This time, 21 is not a root, but the heap property of the tree is not afected by decreasing the key, so all we have to do is to change the value.

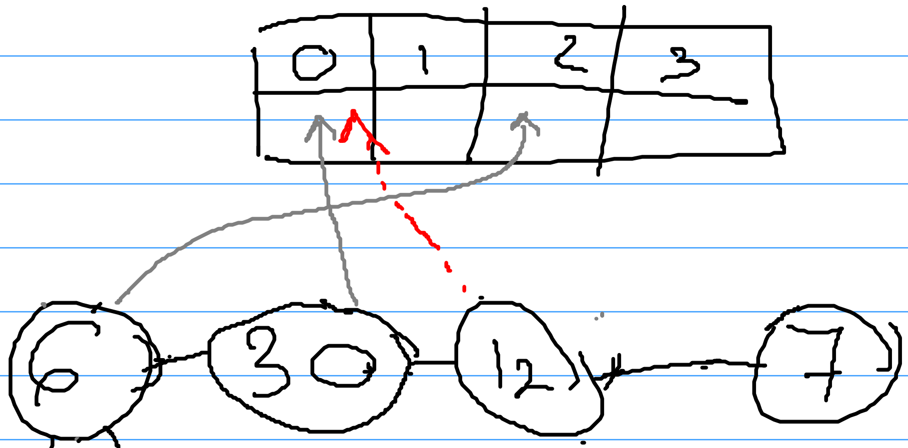
(xi) ExtractMin

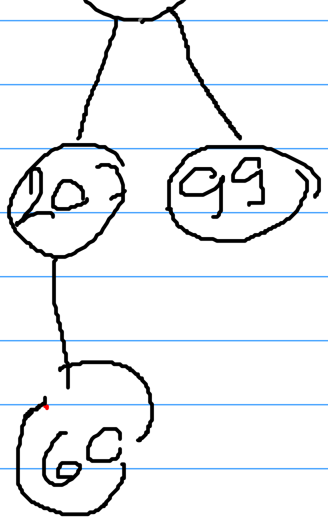
(a) Step 1



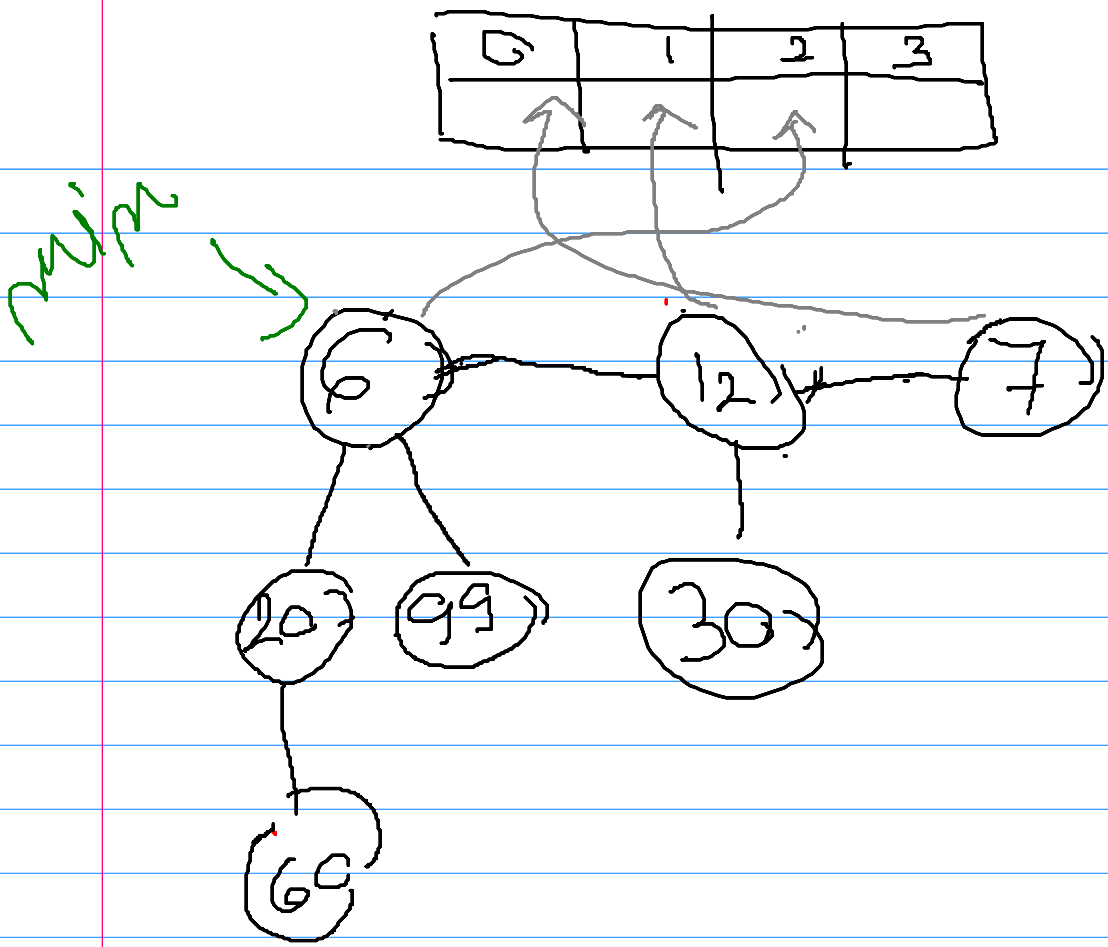


b) Step 2





We have 2 nodes with the same degree, so we have to merge them


0
1
2
3
min
6
12
7
20
99
30
60

1.2 Exercise 2

$$\boldsymbol{\phi = t + 2 \times m}$$

(1) *Why* t *?*
Because every node becomes root at most once.

(2) *Why* $2 \times m$ *?*
For every marked node, we need to perform 2 actions :

1. Cutting
2. Updating t (i.e. we have one more tree, so $t' = t + 1$)

1.3 Exercise 4

If the "peculiar constraint" would not be enforced, we could end up with a node with n descendants having more than $lg(n)$ children. This happens because, for example, we can decrease the key of all the grandchildren of the root such as they violate the heap property. Doing this, we would end up with having a n-node tree with a root and $n - 1$ children.