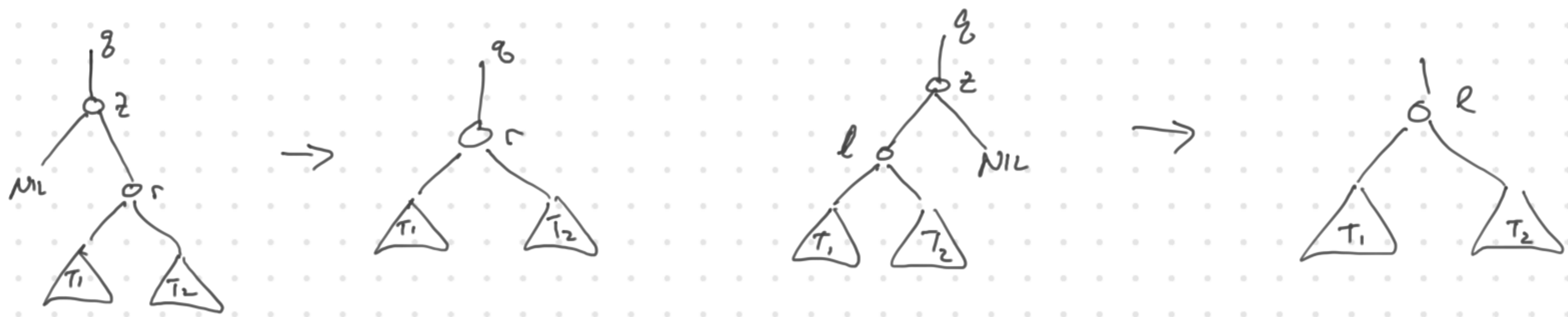
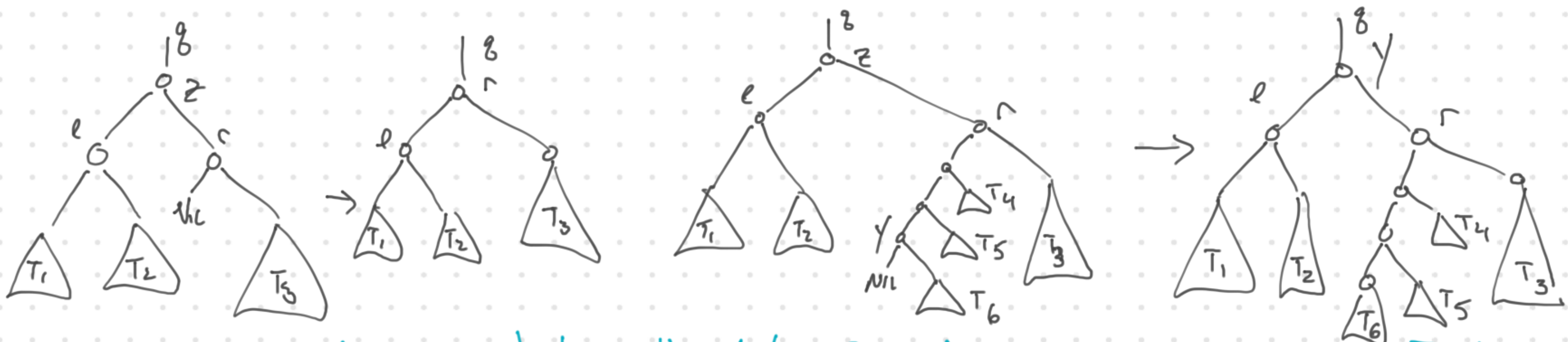
Deletion in Binary trees





we want subroutine which will take 2 nodes u + v + take the $T(v)$
+ glue it into the place of u

Transplant (T, u, v) – replace the subtree rooted at u w/ T(v)

```
if u.parent = Nil
    T.root = v
else if u = u.parent.left
    u.parent.left = v
else  u.parent.right = v
if v ≠ Nil
    v.parent = u.parent
```

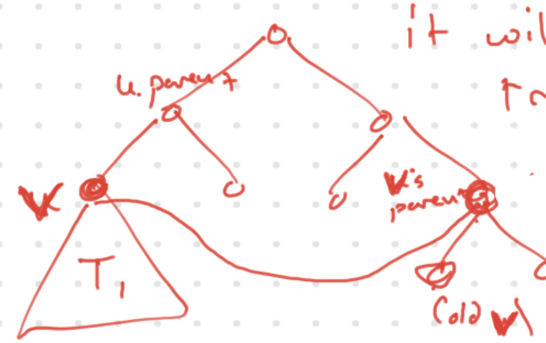
} if u is the root, we replace T w/ T(v)

Note v is not deleted & ~~remains~~ still points to its old children & parent

Does Transplant (T, u, v) always result in a tree? **No**

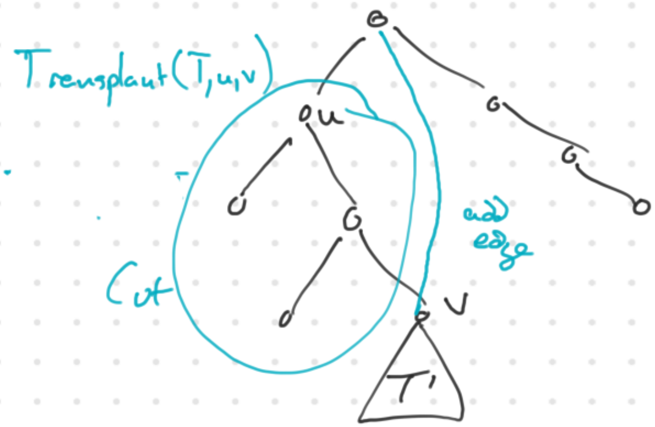
not necessarily, but it will be a tree if

Transplant(T, u, v) →



we can either replace v. parent's ~~child~~ appropriate child w/ NIL

as written, we remain a tree <u>if</u>



if v is a descendent of u ⇒ Transplant yields a tree

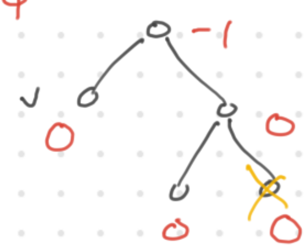
Deletion in a binary tree

```
Tree-delete (T, z)
    if z.left = Nil
        Transplant (T, z, z.right)
    elseif z.right = NIL
        Transplant (T, z, z.left)
    else y = Tree-min (z.right)
        if y = z.right
            Transplant (T, z, y)
            y.left = z.left
        else
            Transplant (T, y, y.right)
            y.left = z.left
            y.right = z.right
            y.parent = z.parent
            if z.parent.right = z
                z.parent.right = y
            else z.parent.left = y
```

Runtime = $O(\text{height}(T))$

note height of the empty tree is defined to be -1

φ



AVL BT ✓

deleting this leaf does leave a AVL BT tree

T - v

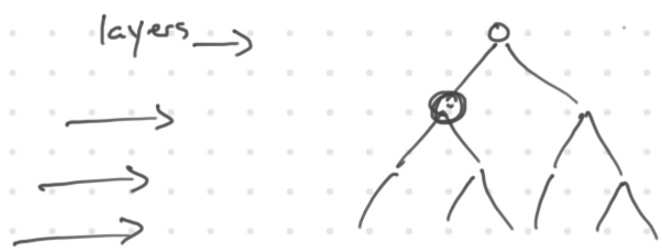
No longer an AVL BT

pf

Let T be an AVL BT of height k on n nodes

$\Rightarrow T$ is a subgraph of the complete binary tree of height k $\Rightarrow n \leq 2^k - 1$

T is also an induced subgraph

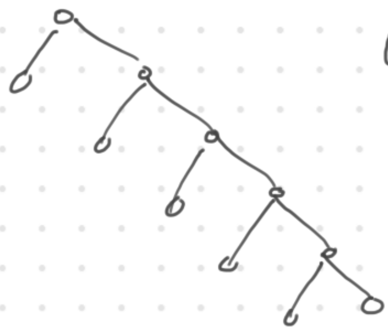


define layers $L_i :=$ vertices at distance i from root

$i = 0, \dots, k$

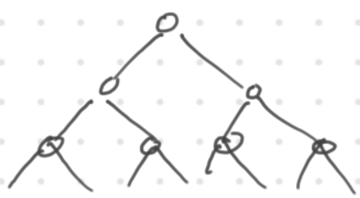
any vertex in L_i, $i \leq k-2$

f(height(T))



height(T) could be $O(n)$

if T is complete binary tree



height $= \log_2 n$

Def load balance function $\varphi(v)$

$\varphi(v) =$ height$(T(v.\text{left}))$ − height$(T(v.\text{right}))$

(note could be (+) or (−))

if T is a complete binary tree Then $\varphi(v) = 0$

<u>Def</u> an AVL Binary tree to be one where $\varphi(v) \in \{-1, 0, 1\}$ $\forall$ vertices v.

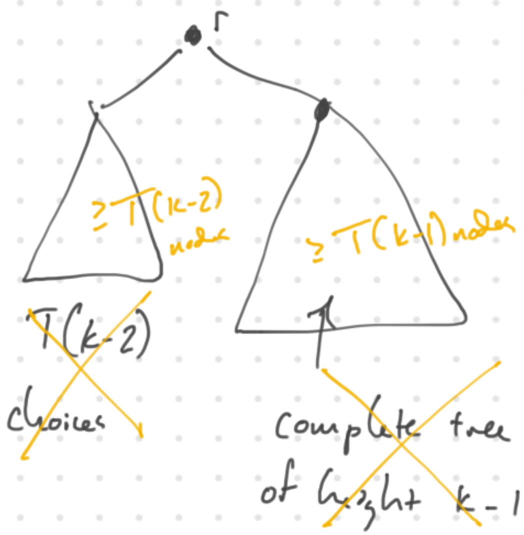
<u>Prop</u> height of an AVL BT on n nodes is $O(\log n)$.

$T(k) \geq T(k-1)$

Def $T(k) :=$ min # of nodes in an AVL BT of height k

$T(0) = 1$
$T(1) = 2$

Cl $T(k) \geq T(k-1) + T(k-2)$



AVL BT of height k

$T - r$ is two AVL trees either both of height $k-1$ or one of height $k-2$ + one of height $k-1$

so there are at least $T(k-1) + T(k-2)$ nodes in an AVL trees of height k

$\Rightarrow T(k) \geq k^{th}$ fibbonacci #

$\Rightarrow T(k)$ which is exponential in k.

choices are of distinct AVL trees of height $k-1$

exponential function in k $\le$ # of nodes in tree $\le 2^k$

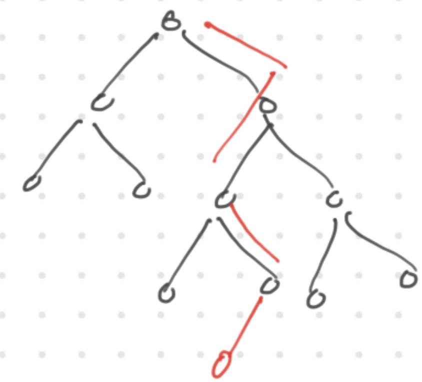
$\Rightarrow$ a AVL BT on n nodes has $O(\log n)$ height.

How to maintain an AVL-BT doing insertions + deletions

C1 Let T' be obtained from T by inserting a vertex v. Then

$$\varphi_T(v) - 1 \le \varphi_{T'}(v) \le \varphi_T(v) + 1$$

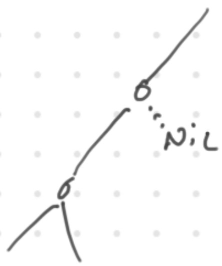
Pf

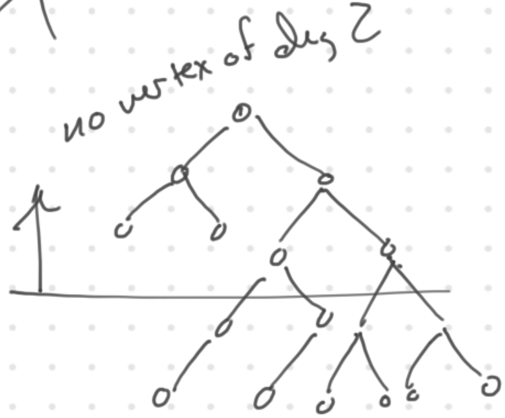


a new vertex is always added to a leaf $\Rightarrow$ $\text{height}_{T'}$ of any vertex changes by ≤ 1

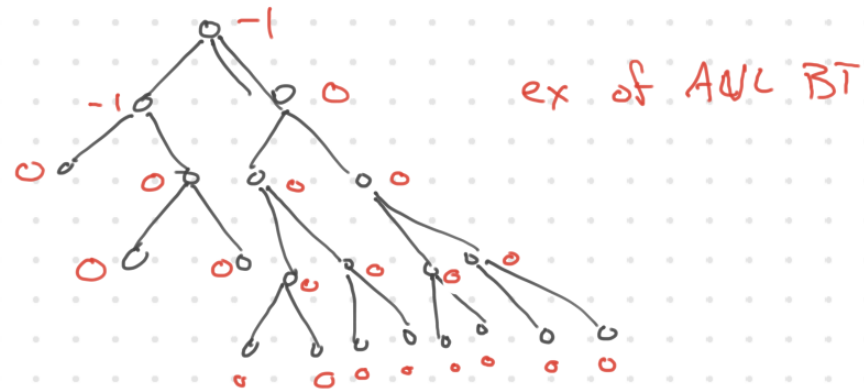
$\Rightarrow$ $\varphi_{T'}(v)$ changed by at most 1 must have deg = 3

if not





C1 it's possible That T has leaves not in The bottom two layers



let $T(k) :=$ min # of nodes in a AVL BT of height k

$T(0) = 0$

$T(1) = 1$

C1 upon deletion, $\varphi(v)$ changes by at most 1 ie if T' obtained by a deletion

$$\varphi_T(v)-1 \le \varphi_{T'}(v) \le \varphi_T(v)+1$$

pf in each of the 4 cases, the new tree T' is obtained from the old tree by contracting an edge ⇒ height of any vertex in T' differs by at most 1 from height in T ⇒

$$\varphi_T(v)-1 \le \varphi_{T'}(v) \le \varphi_T(v)+1$$

Conclusion: after inserting / deleting a vertex in an AVL BT, we just need to fix nodes w/ $\varphi(v)=2$ or $\varphi(v)=-2$

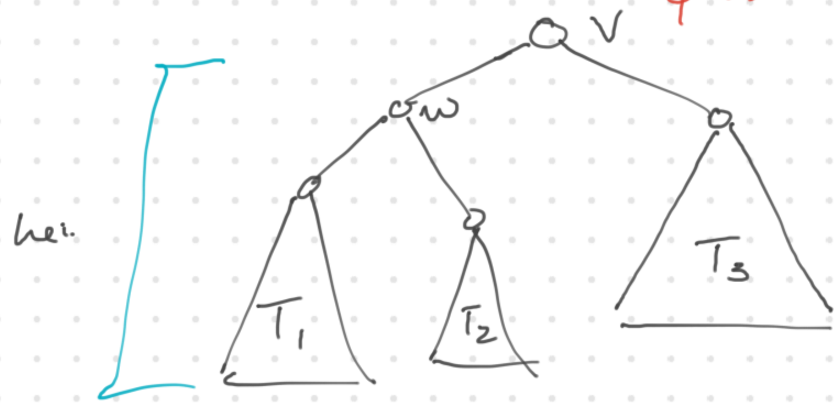
Fixing vertices w/ $\varphi=2$

1) if $\exists$ more than one vertex w/ $\varphi(v)=2$, ~~let~~ Fix v to be such a vertex as far from root as possible

Ex prove that all other nodes x w/ $\varphi(x)=2$ are ancestors of v.

CASE

$\varphi(v)=2$

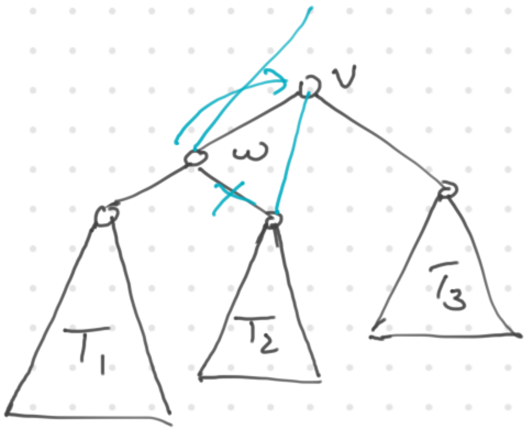


height w = height $(T_3)+2$

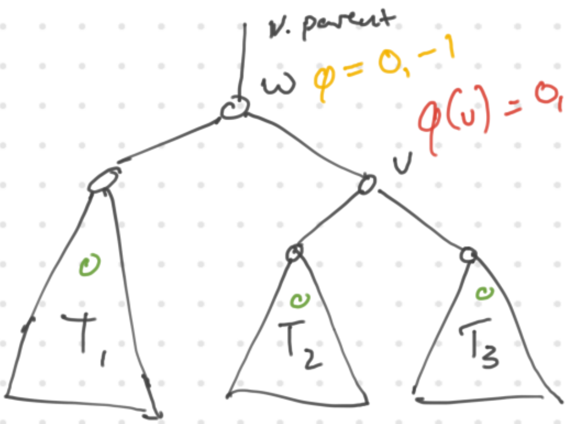
$\Rightarrow$ max (height T_1, height T_2) = height $T_3 + 1$

we split the cases depending on whether T_1 or T_2 has height height(T_3) +1

<u>1st Case</u> height (T_1) = height(T_3)+1



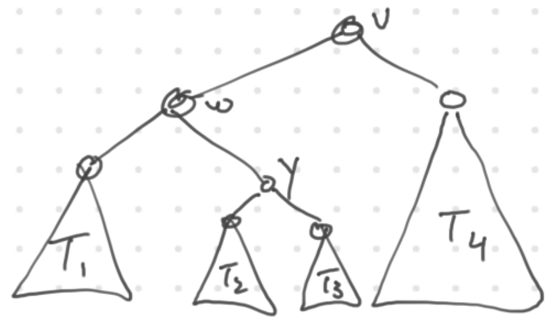
R-rotation



any node in T_i has same φ value

you can also show that any node above v w/ $\varphi = 2$ is now fixed.

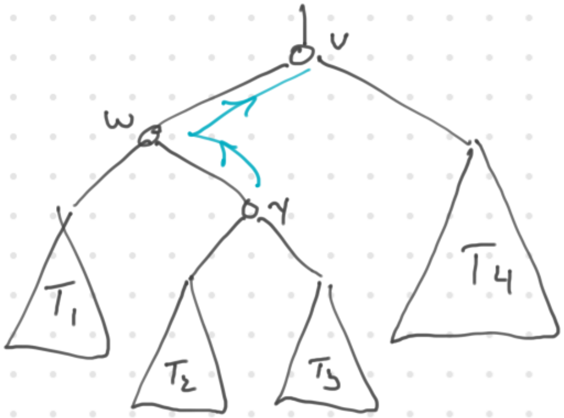
The other case is



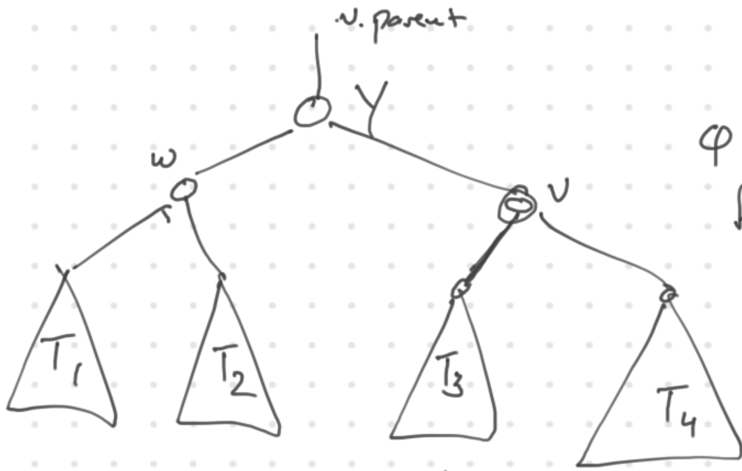
height(w) = height (T_4)+2

+ height(y) = height (T_4) + 1

ie one of T_2 or T_3 has height = height (T_4)



LR-rotation



φ = 0, 1 or -1 for all vertices here.