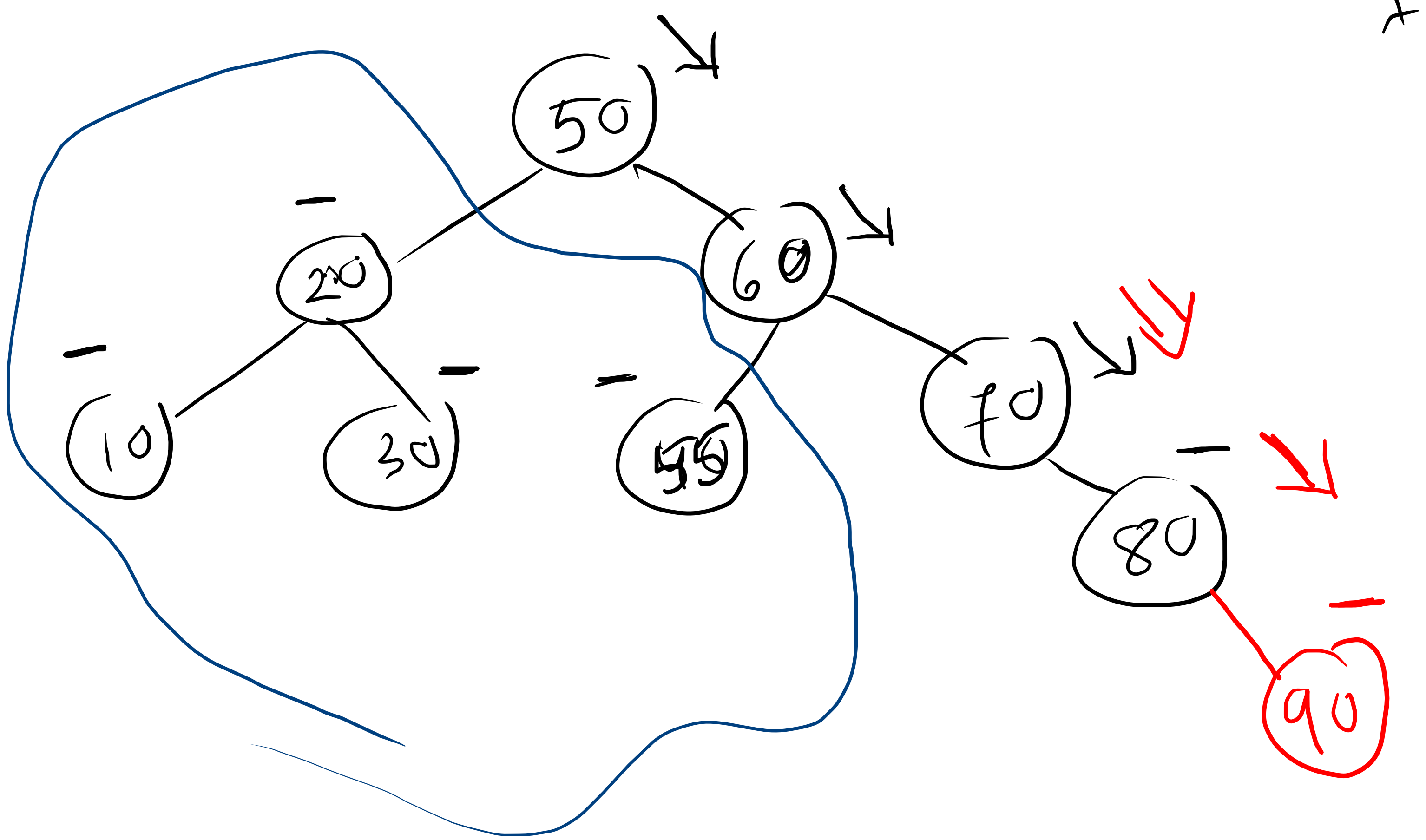
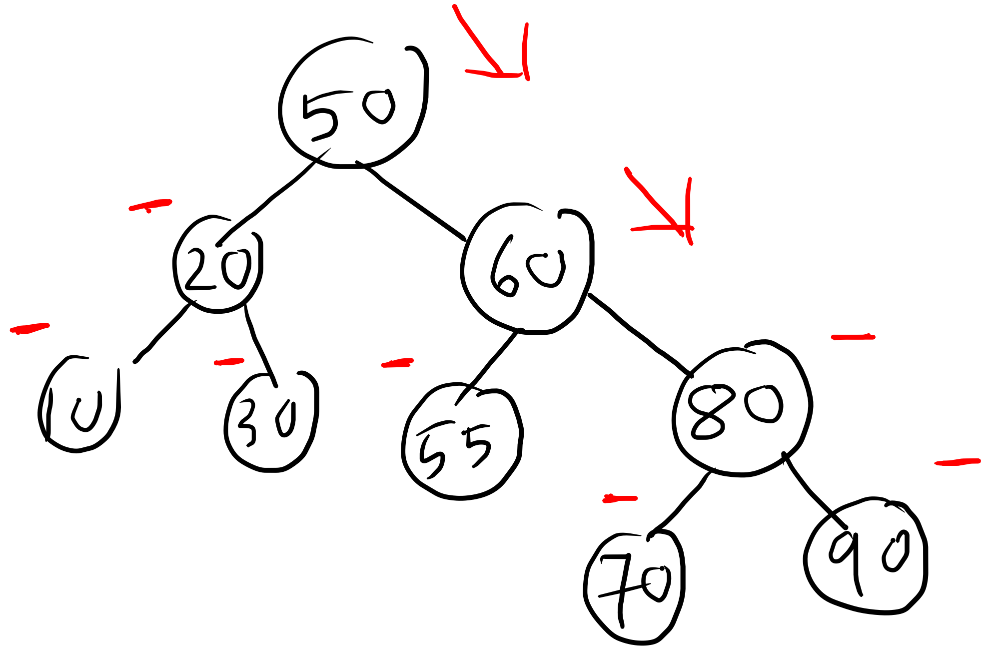
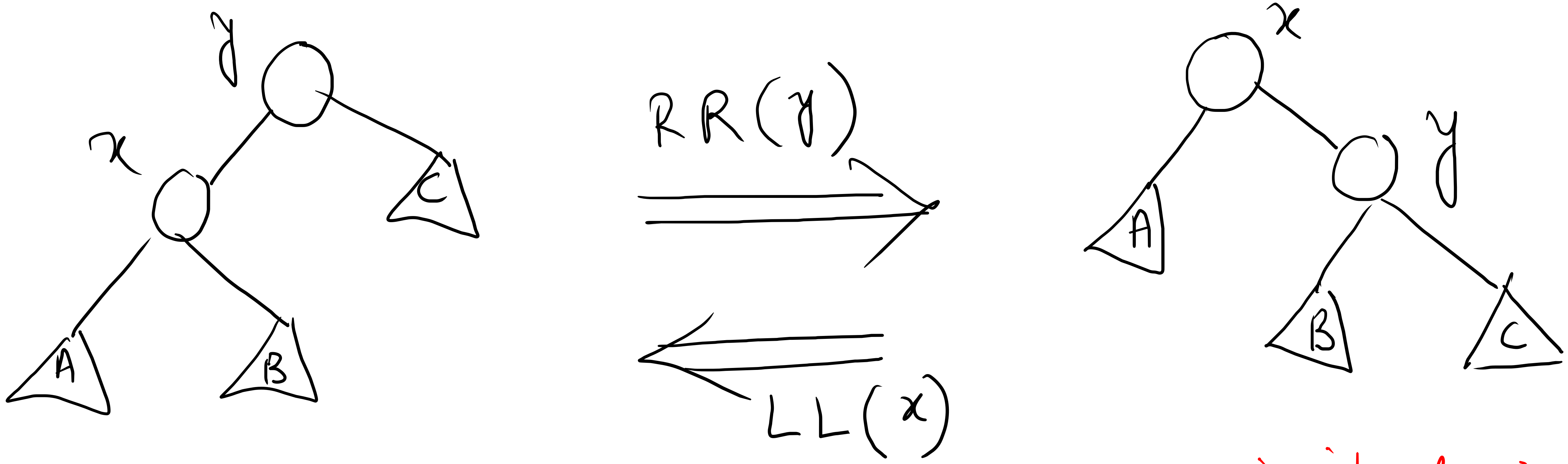
AVL tree insertion

Insert 90



$\Rightarrow$ LL(70)





Similarly we have, LL(x)

RR(y)

x = y.left

tmp = x.right

x.right = y

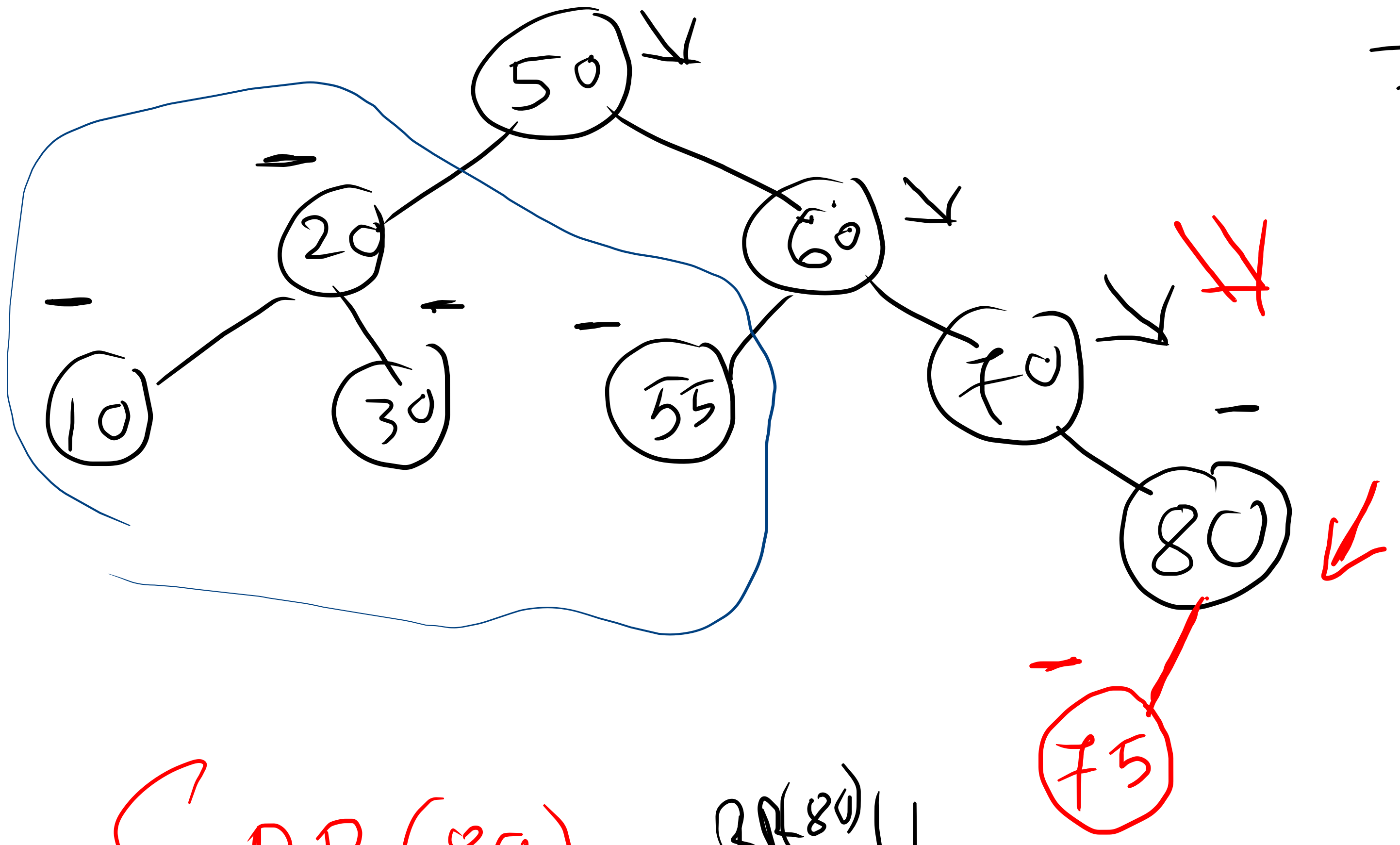
y.left = tmp

y.height = max{height(y.left), height(y.right)} + 1

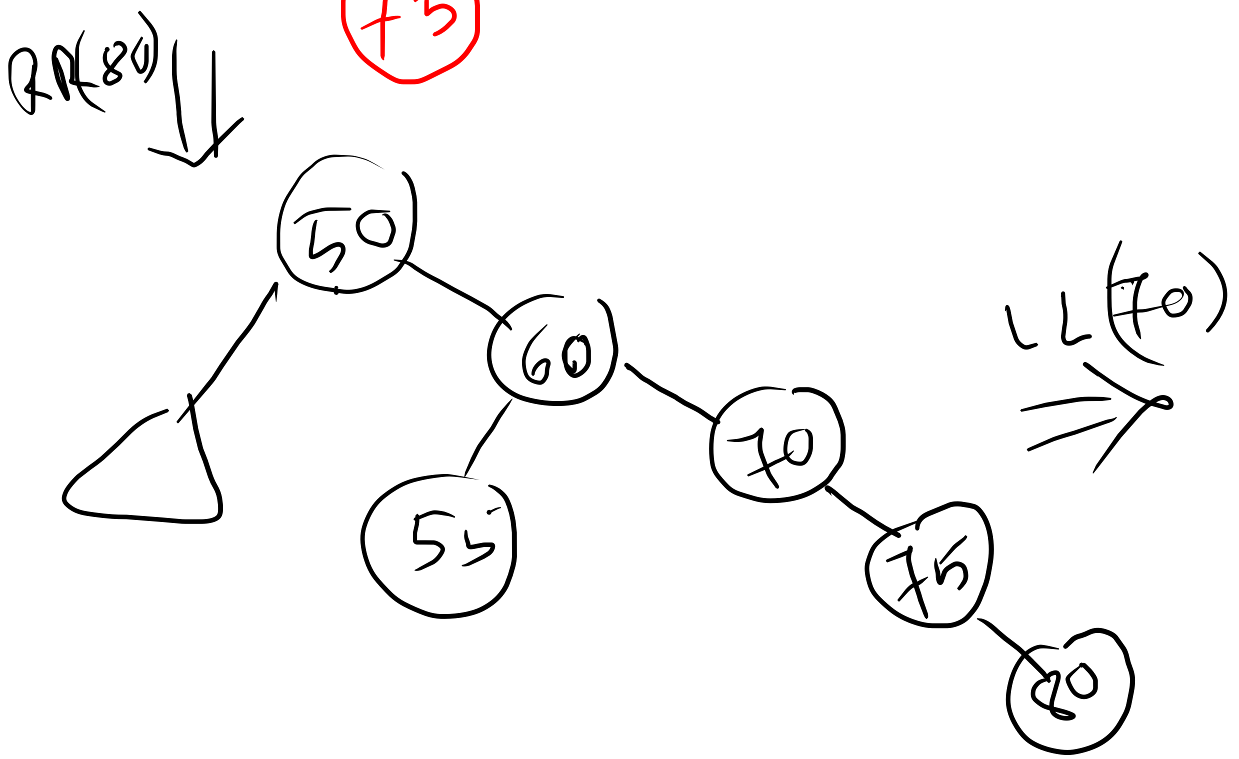
x.height = max{height(x.left), height(x.right)} + 1

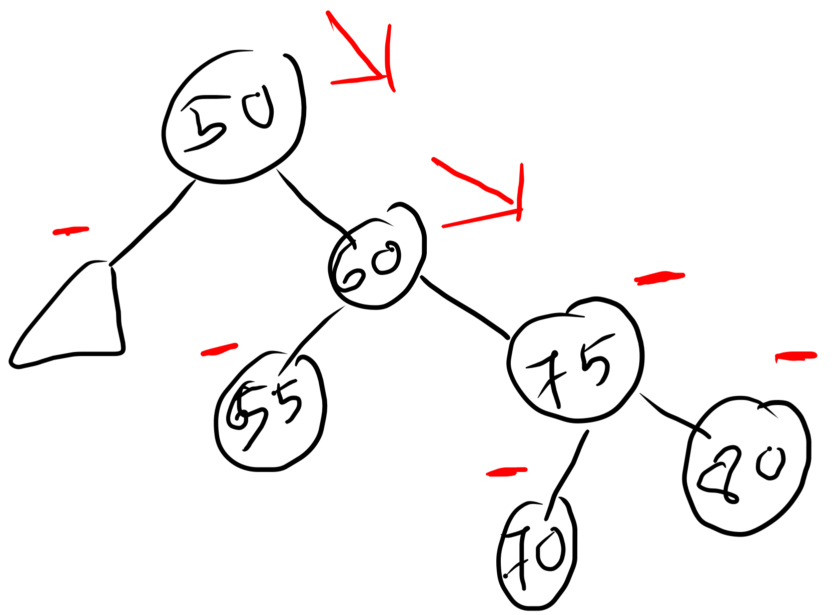
return x.

Insert 75

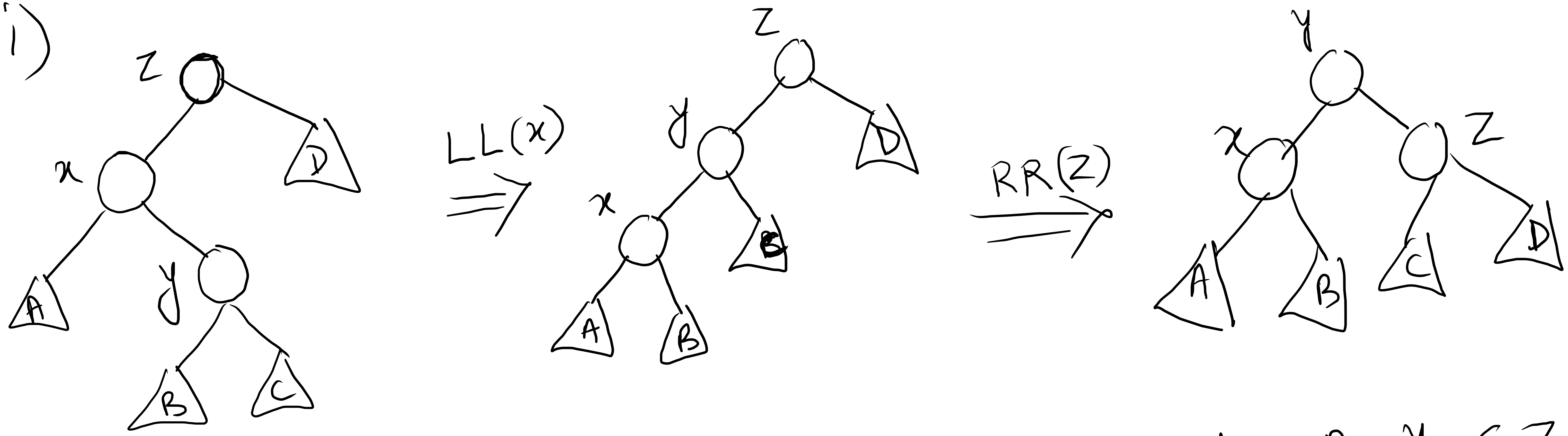


RR(80)
then
LL(70)





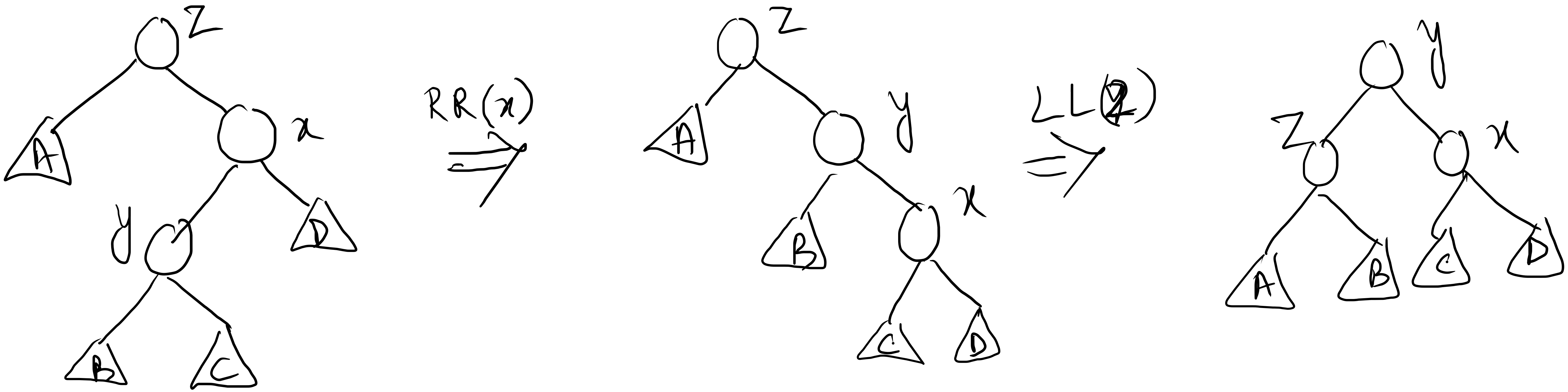
i)



A x B y C Z D

A x B y C Z D

ii)



AVLinsert(T, K)

if T == Null

create and return the node with key K as root.

if K < T.key

T.left = AVLinsert(T.left, K)

else if K > T.key

T.right = ALLinsert (T.right, K)

else

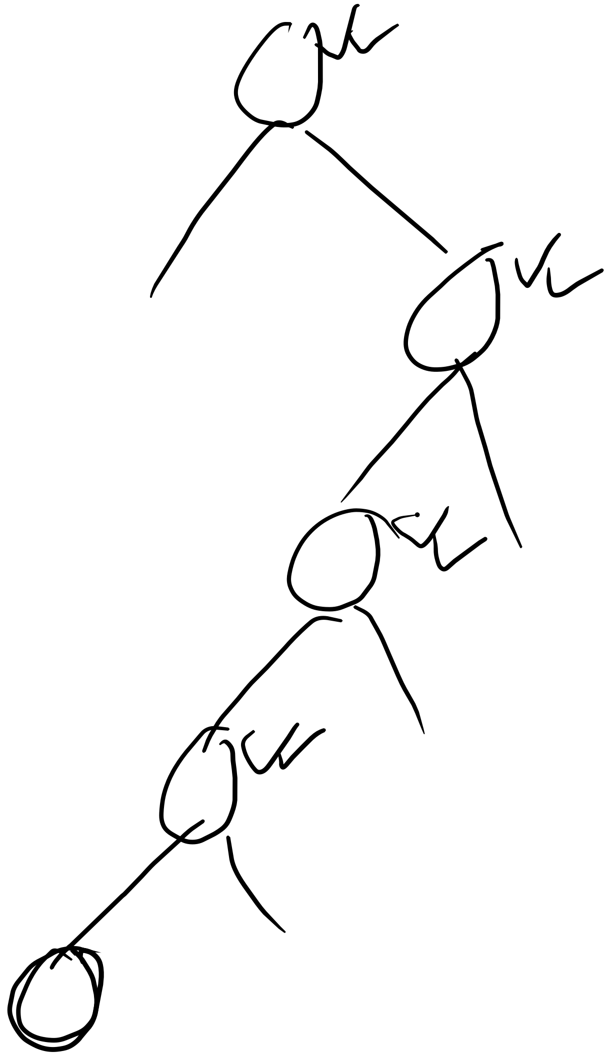
return T

T.height = max{height(T.left), height(T.right)}+1

bl = balance(T)

balance(Z)

return height(Z.left) - height(Z.right)



T

if bl > 1 // double left heavy.

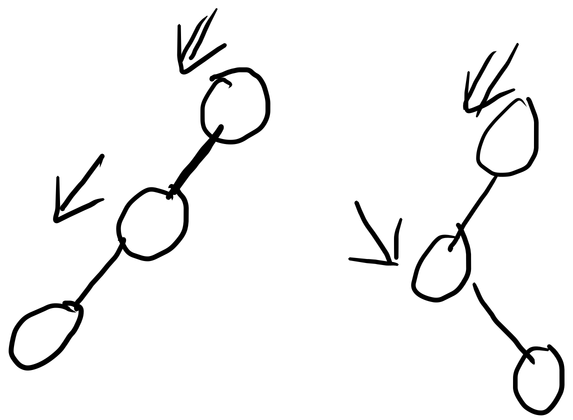
if balance (T.left) ≥ 0 // T.left is left heavy or balance.

RR(T) // single rotation.

else if balance(T.left) < 0 // Right heavy.

LL (T.left)

RR (T) // double rotation.



if bl < -1

if balance (T.right) ≤ 0

LL (T)

else if balance (T.right) > 0

RR (T.right)

LL (T)

return T

Running time:

$O(\log n)$