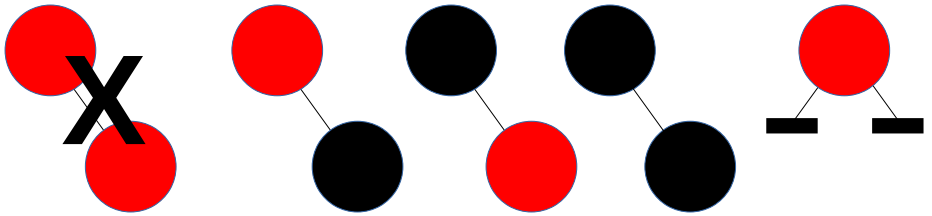
RED-BLACK TREE

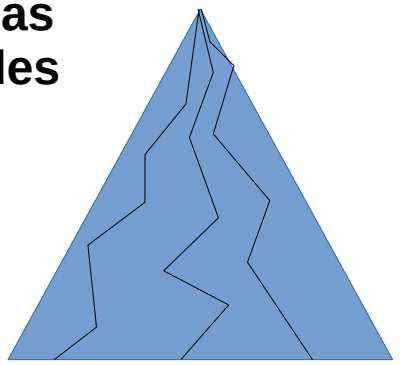
1. **Each node is either Red or Black**
   (null pointers considered black)

2. **Root is always Black**

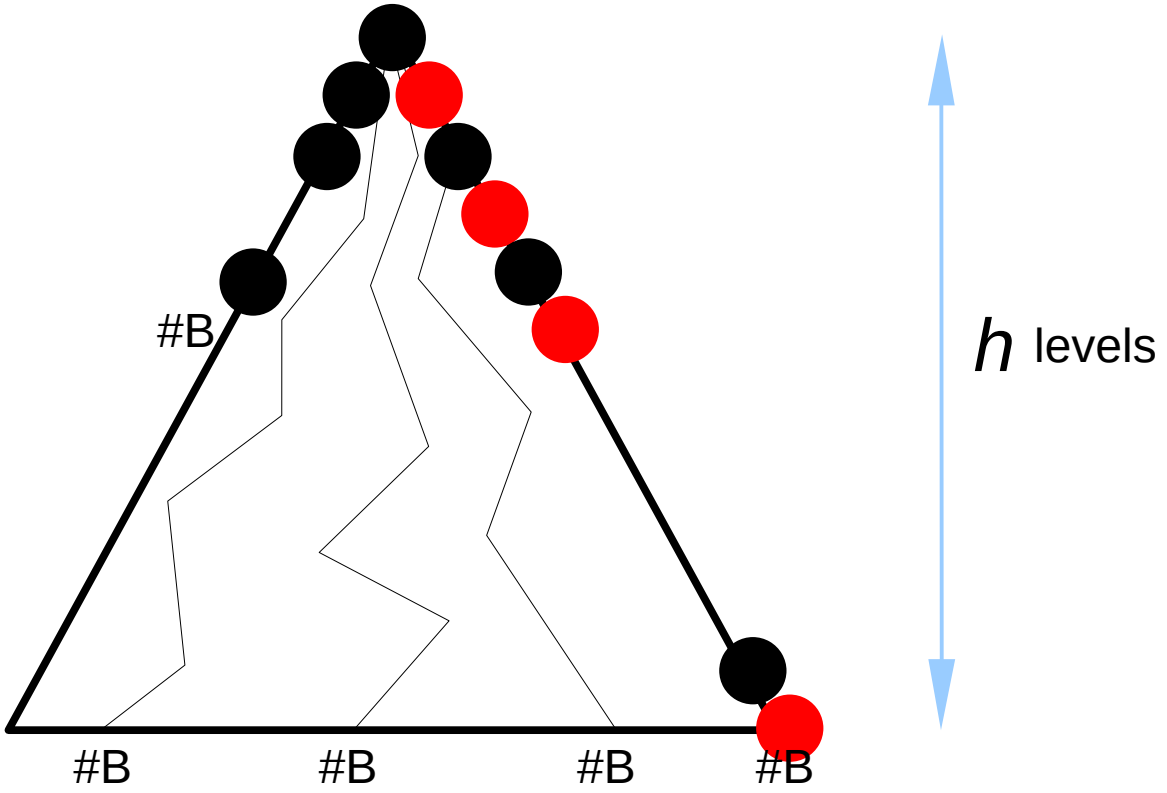
3. **Red nodes must have Black children**
   (cannot have adjacent Red nodes)

4. **Every path from Root to Leaf has**
   the same number of Black nodes





#B = #B = #B



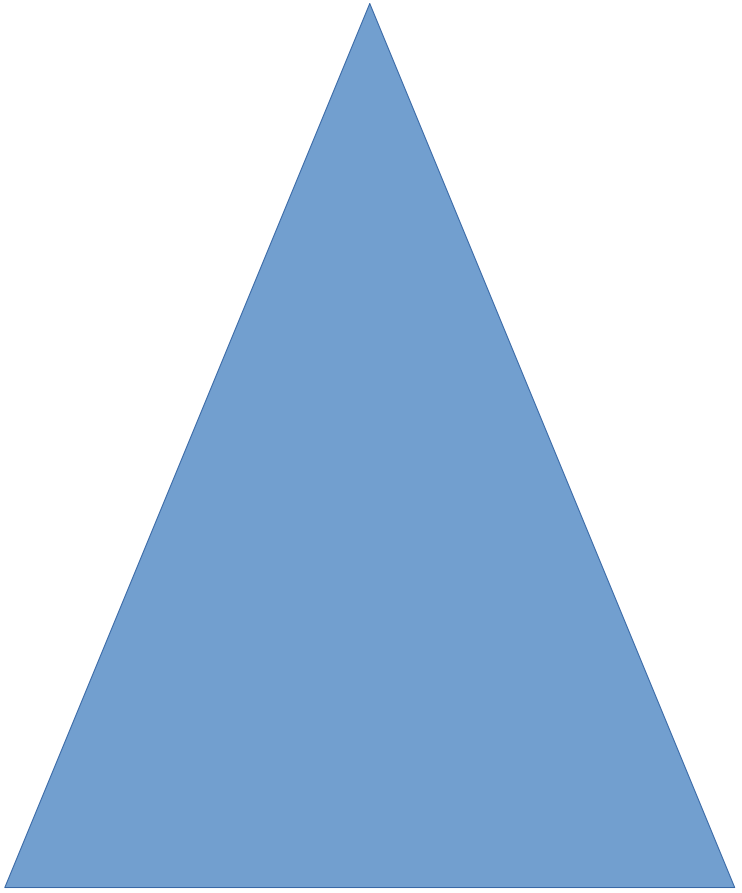
(a) h >= #B — need at least as many levels as number of Black #B

(b) h <= 2#B — longest possible path has alternating Black, Red starting with Black (the root) #B = #R

we get h/2 <= #B <= h

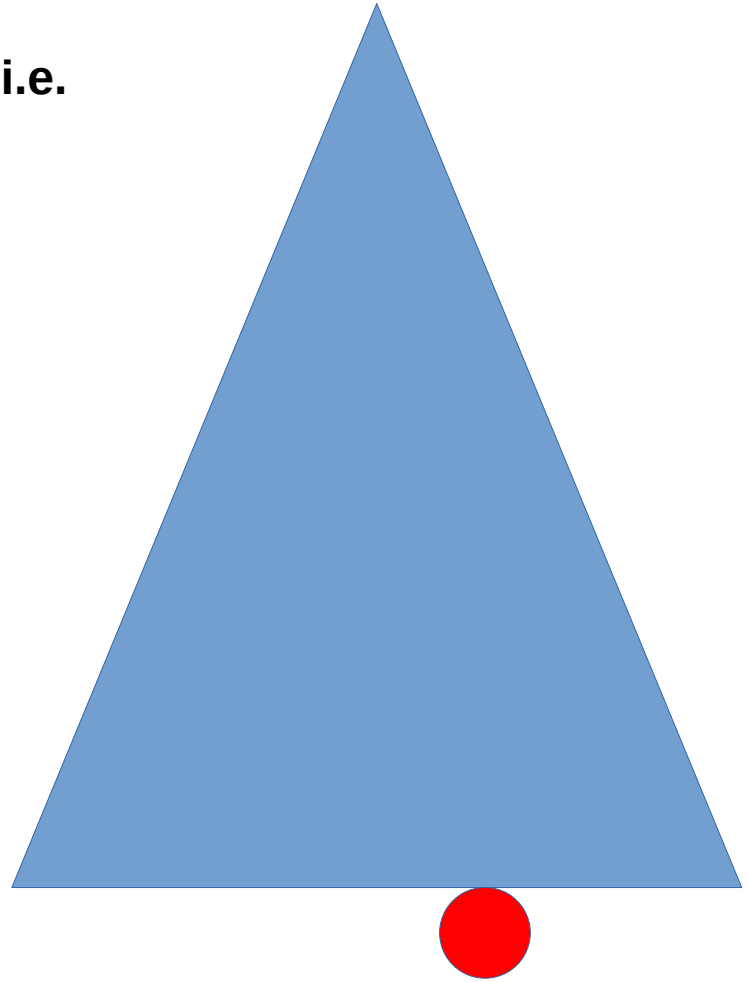
(b) (a)

INSERTION MAIN IDEA



INSERTION MAIN IDEA

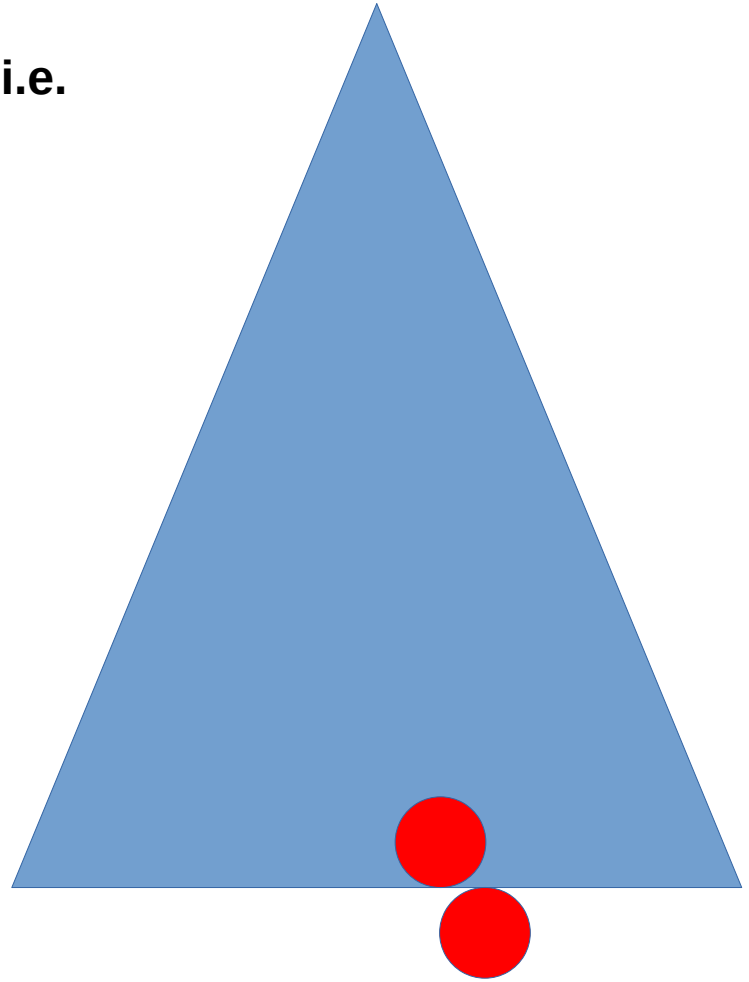
Insert as in regular BST, i.e. as leaf at the bottom



INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

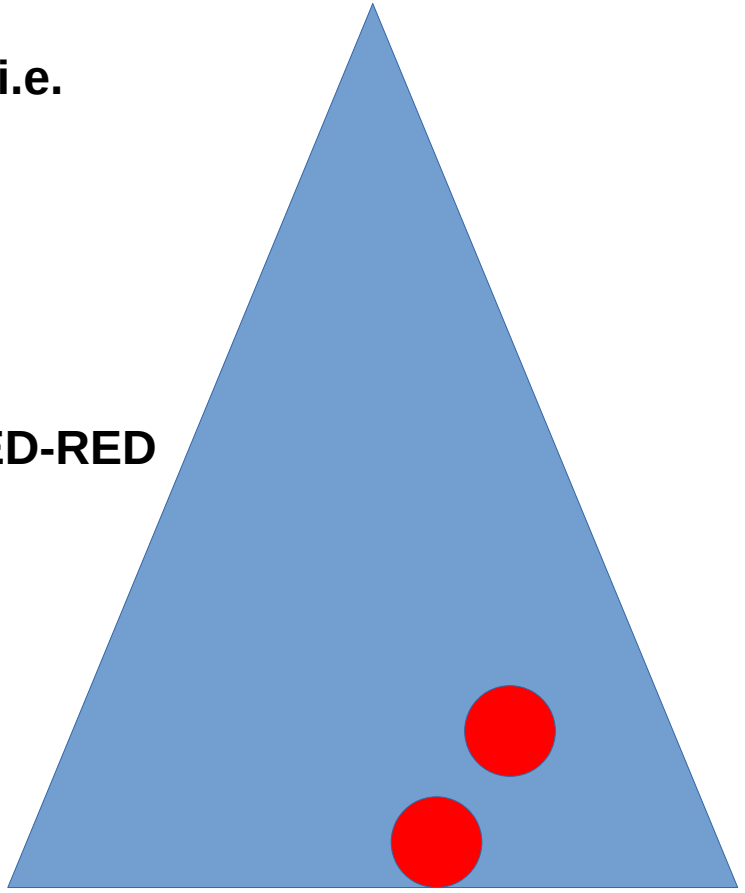


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

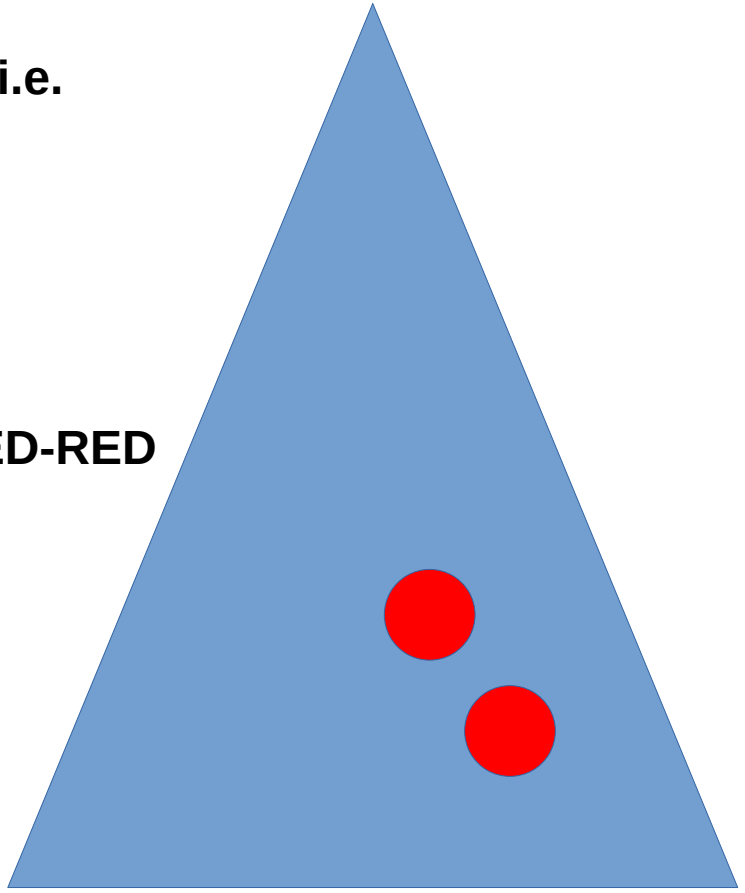


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

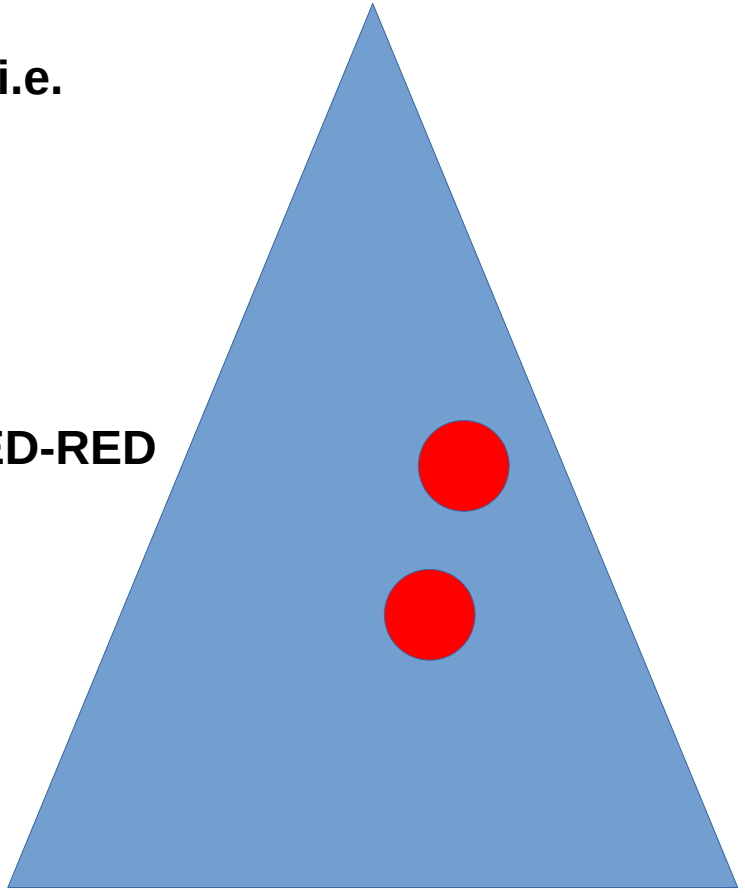


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

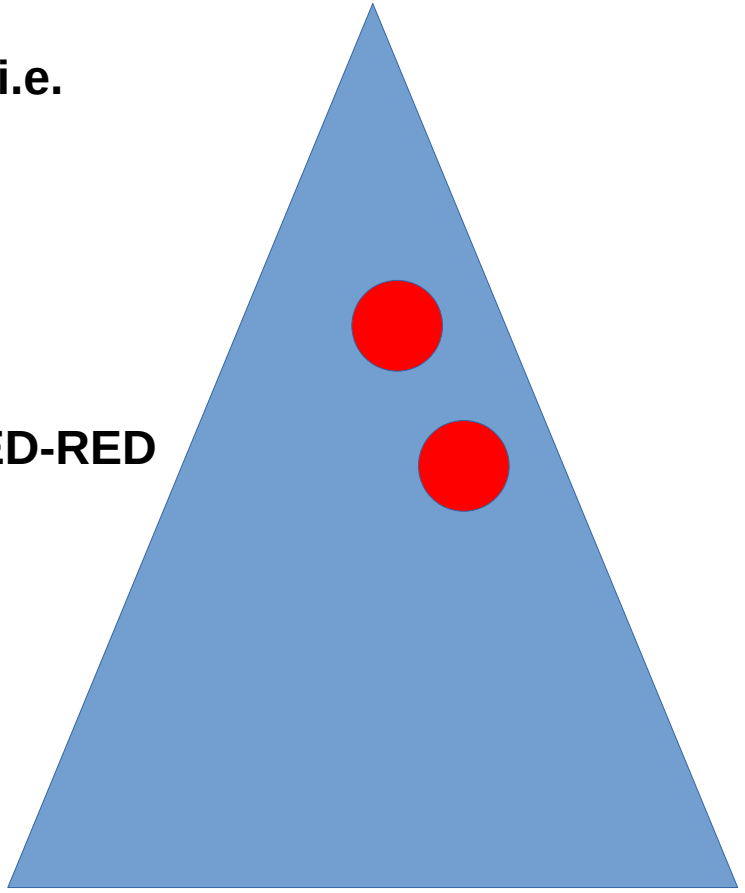


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

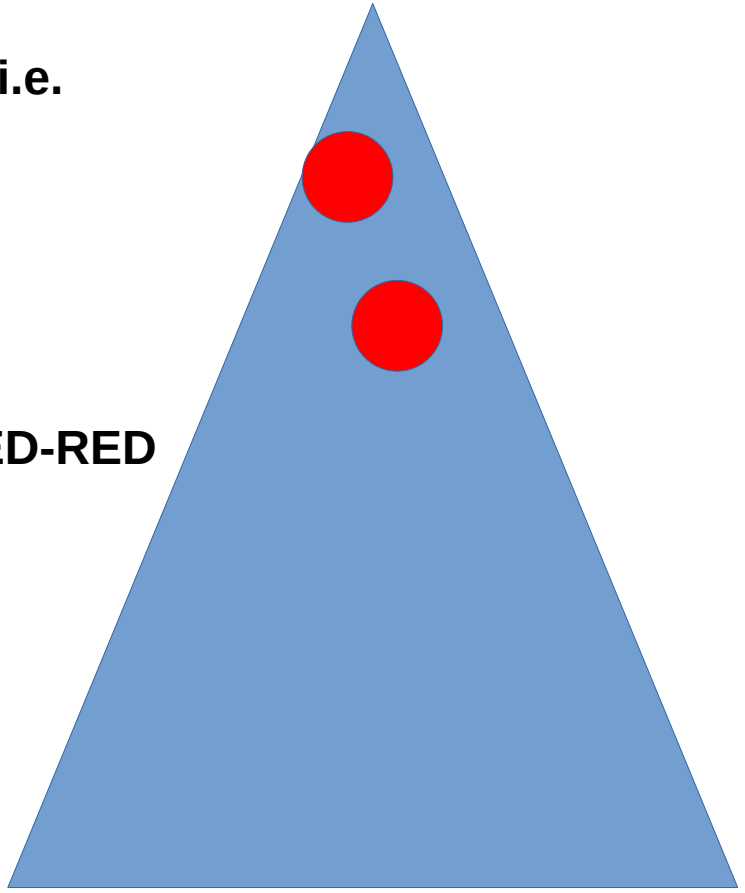


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

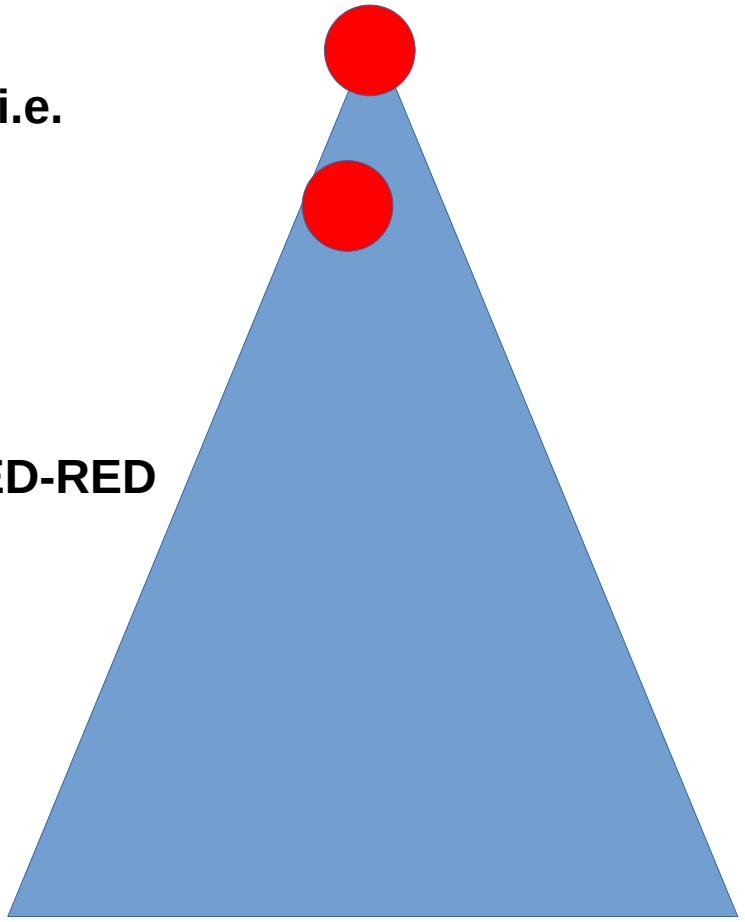


INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up



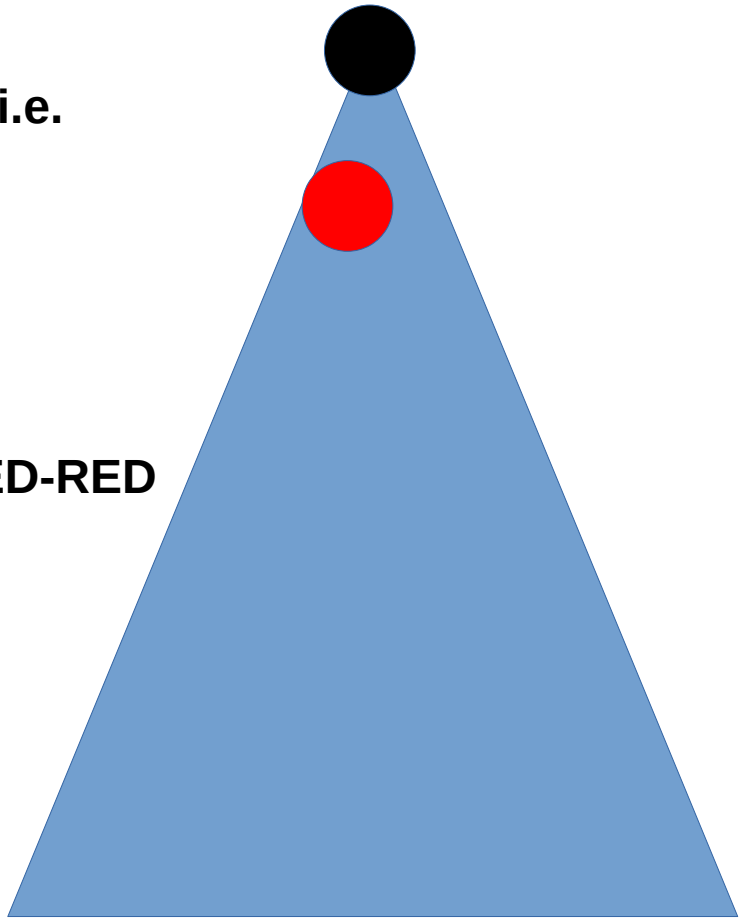
INSERTION MAIN IDEA

Insert as in regular BST, i.e. as leaf at the bottom

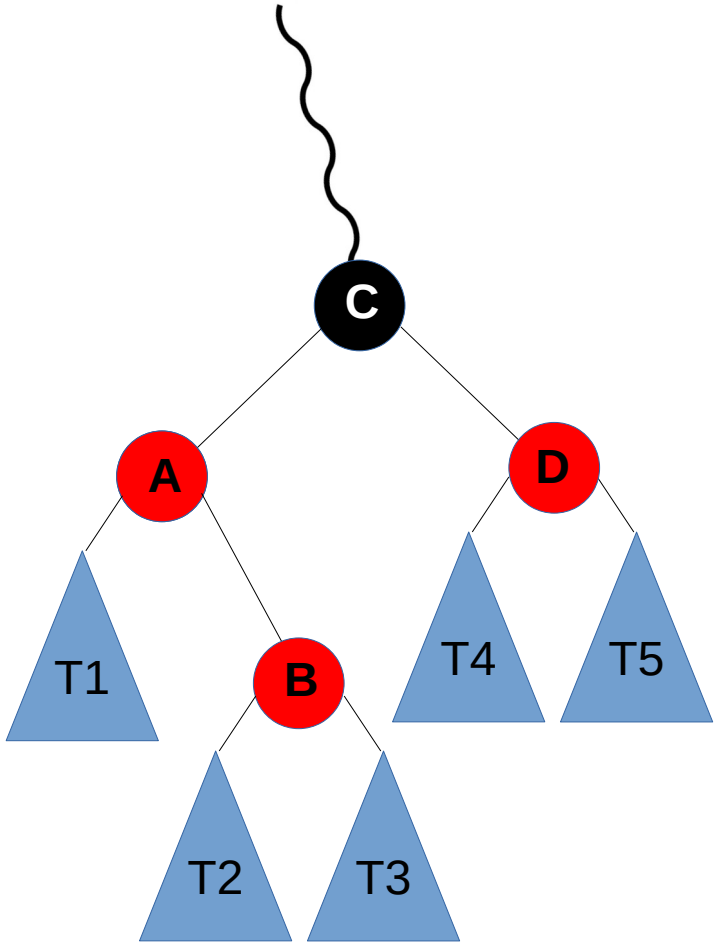
Might create RED-RED violation above

Repair by moving the RED-RED Violation one level up

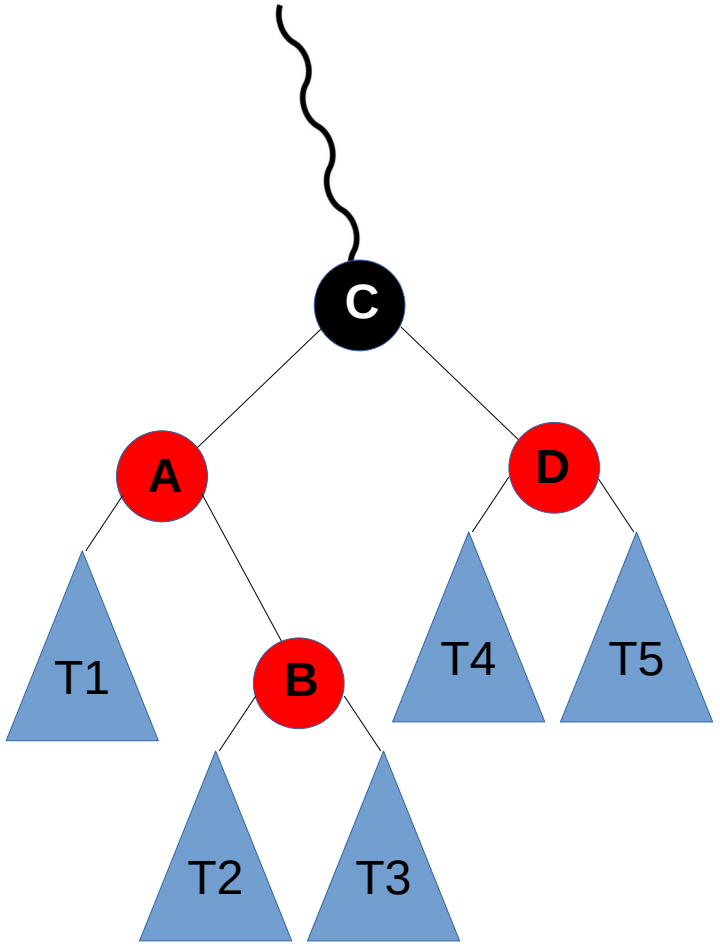
At the end may need to color root BLACK



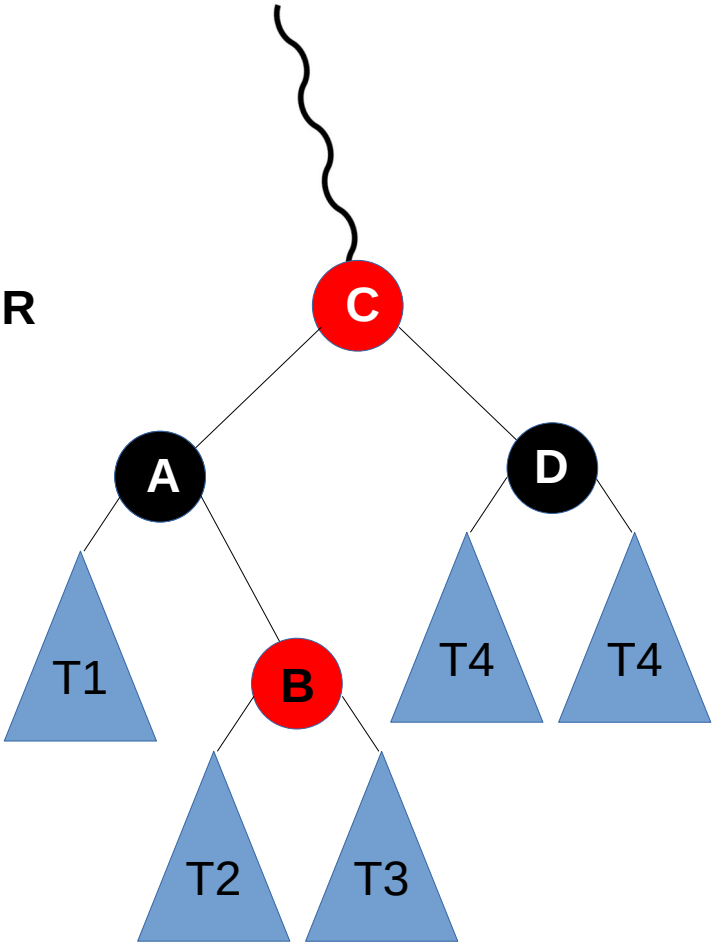
CASE I VIOLATION: RED PARENT, RED UNCLE



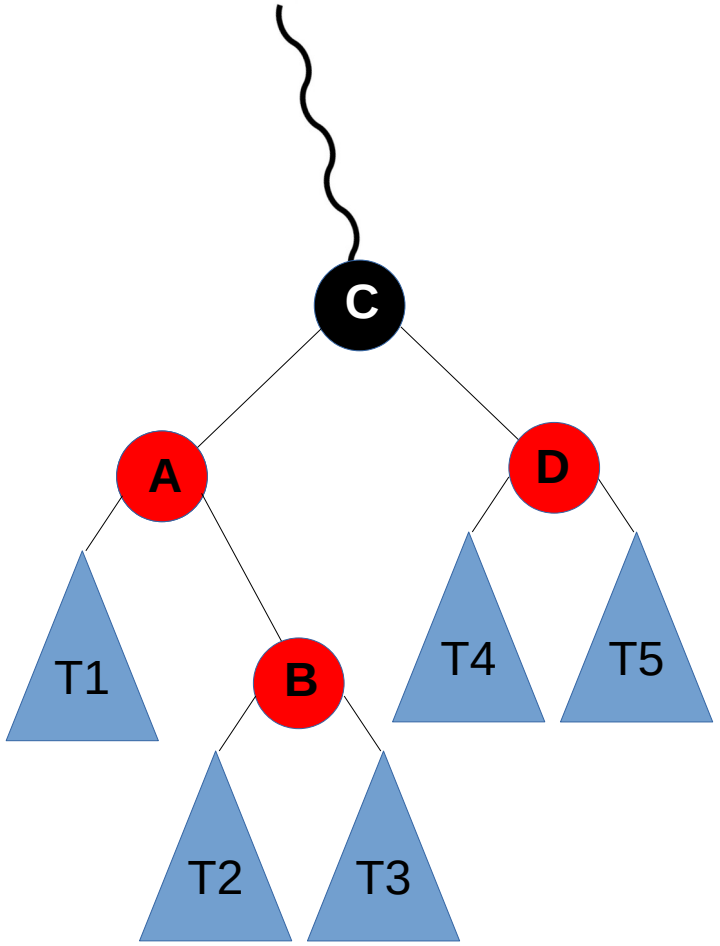
CASE I VIOLATION: RED PARENT, RED UNCLE



RECOLOR

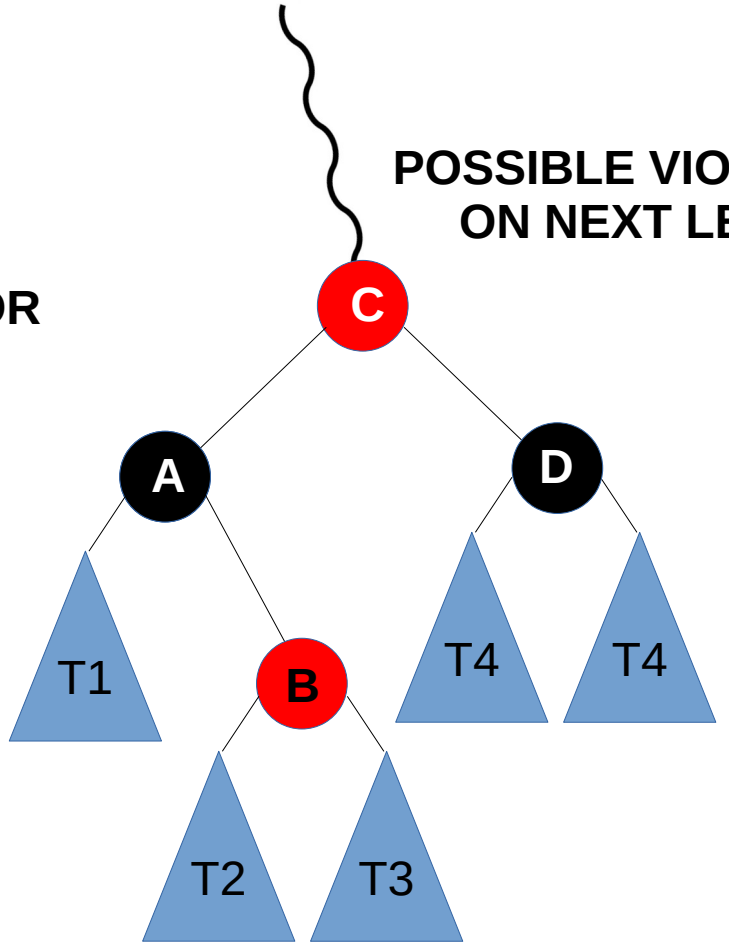


CASE I VIOLATION: RED PARENT, RED UNCLE

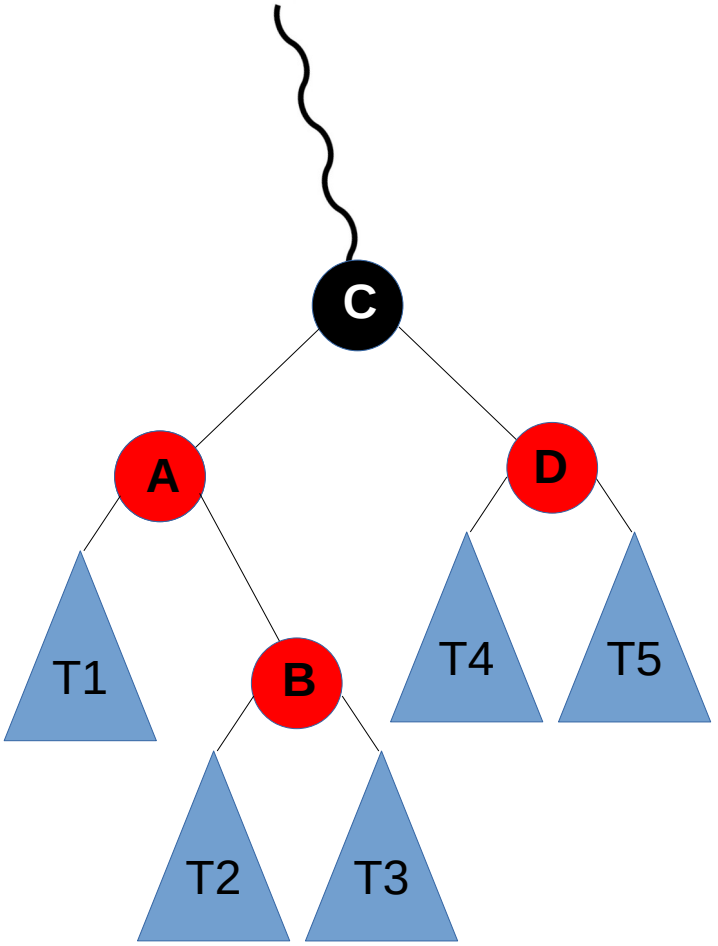


RECOLOR

POSSIBLE VIOLATION ON NEXT LEVEL

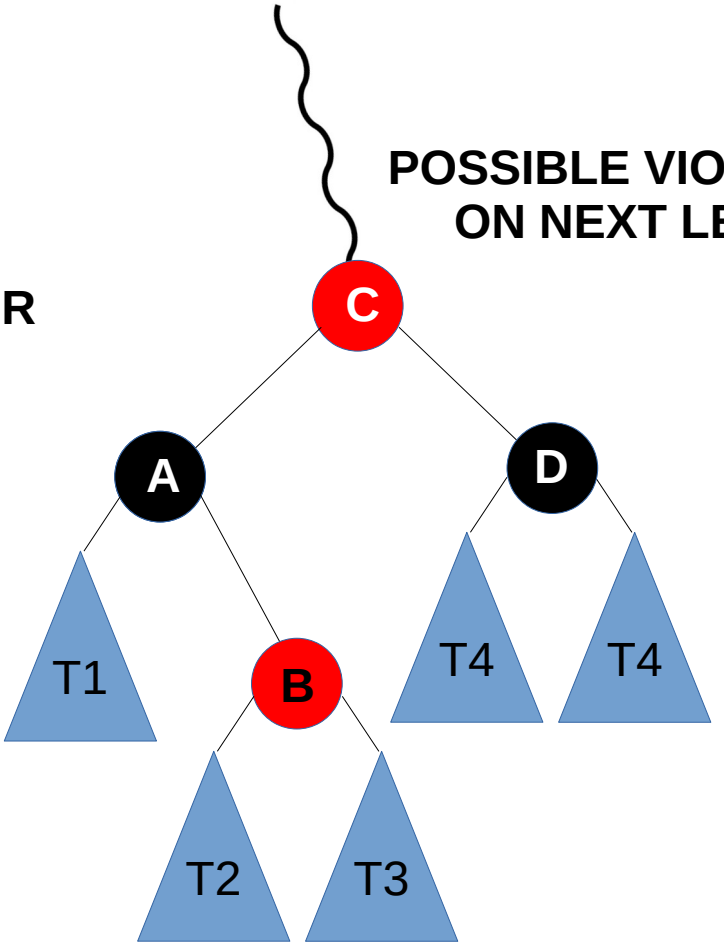


CASE I VIOLATION: RED PARENT, RED UNCLE



RECOLOR

POSSIBLE VIOLATION
ON NEXT LEVEL



Started with:

$\#B = \#B^{top} + \#B^{T1}$
$= \#B^{top} + \#B^{T2}$
…
$= \#B^{top} + \#B^{T5}$

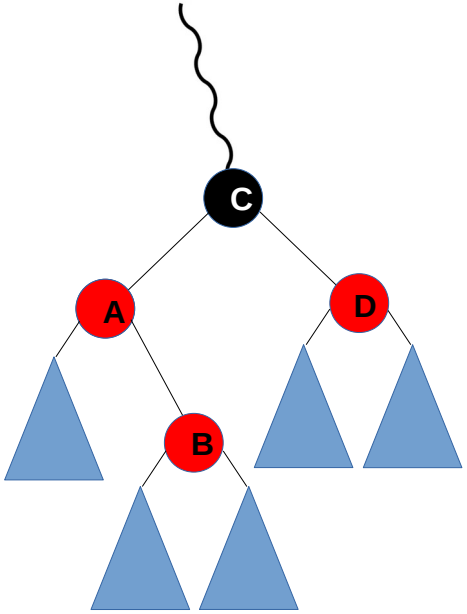
Finished with the same:

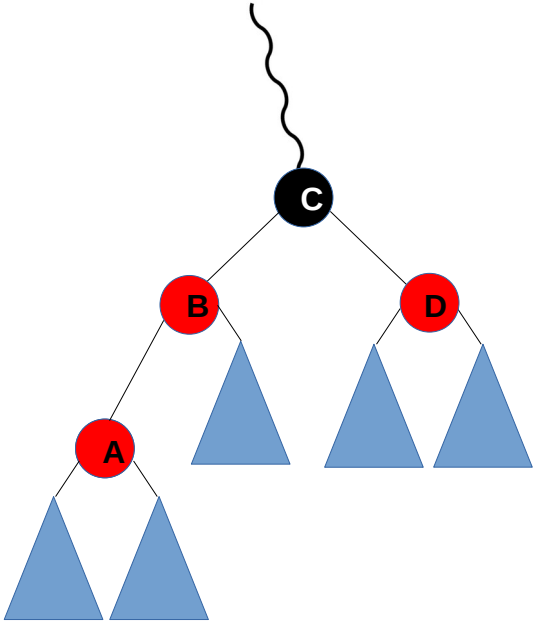
$\#B = \#B^{top} - 1 + 1 + \#B^{T1}$
$= \#B^{top} - 1 + 1 + \#B^{T2}$
…
$= \#B^{top} - 1 + 1 + \#B^{T5}$

on each path
lost 1 for C
but gained 1 for A/D

$\#B^{Ti}$ is the <u>length</u> of a Black path in Ti <u>not total number</u> of Black nodes in Ti

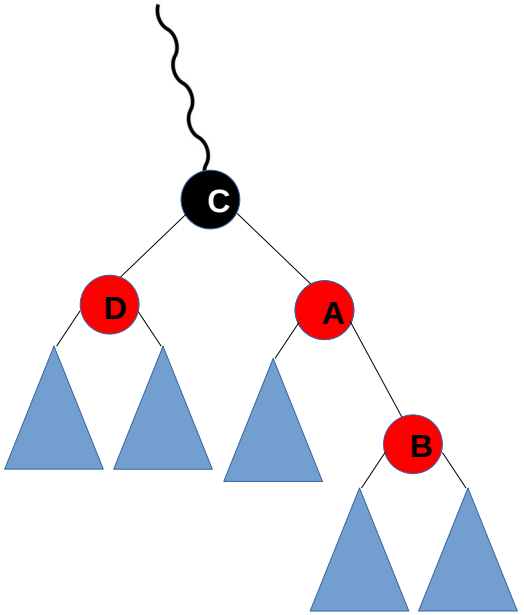
CASE I VIOLATION: RED PARENT, RED UNCLE

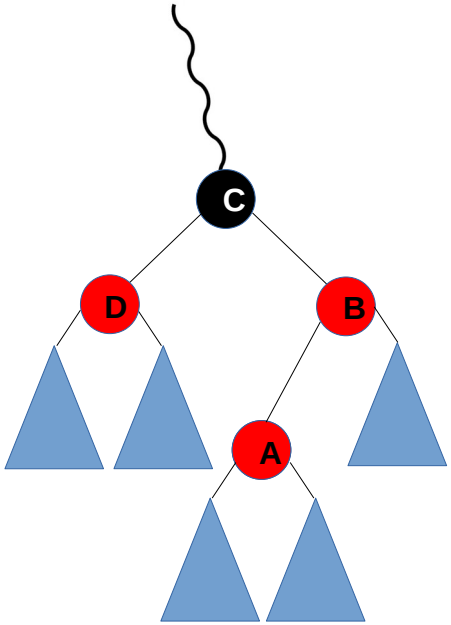




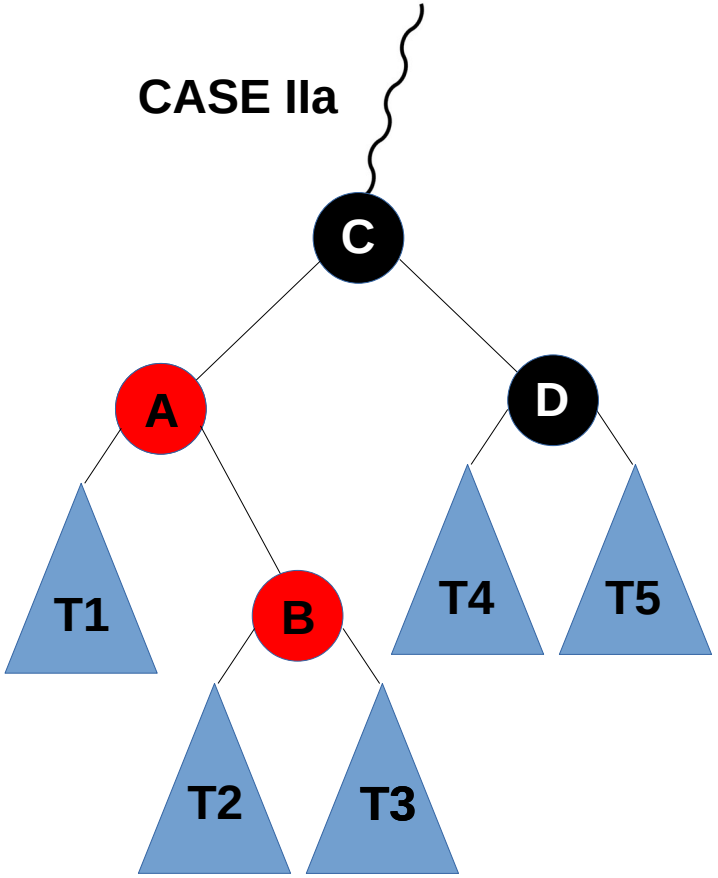
CASE I VARIATIONS

RECOLOR node, uncle, parent

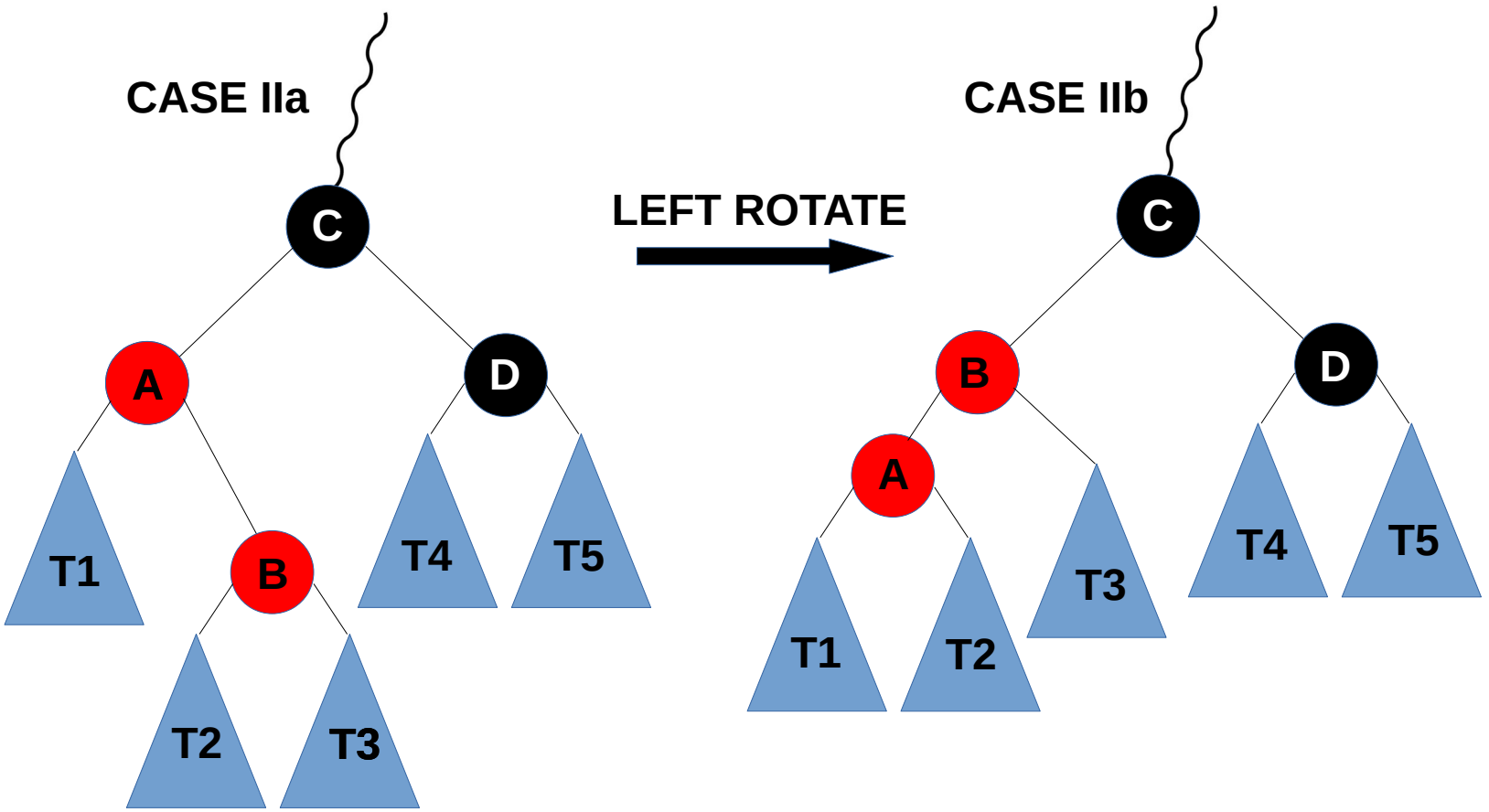


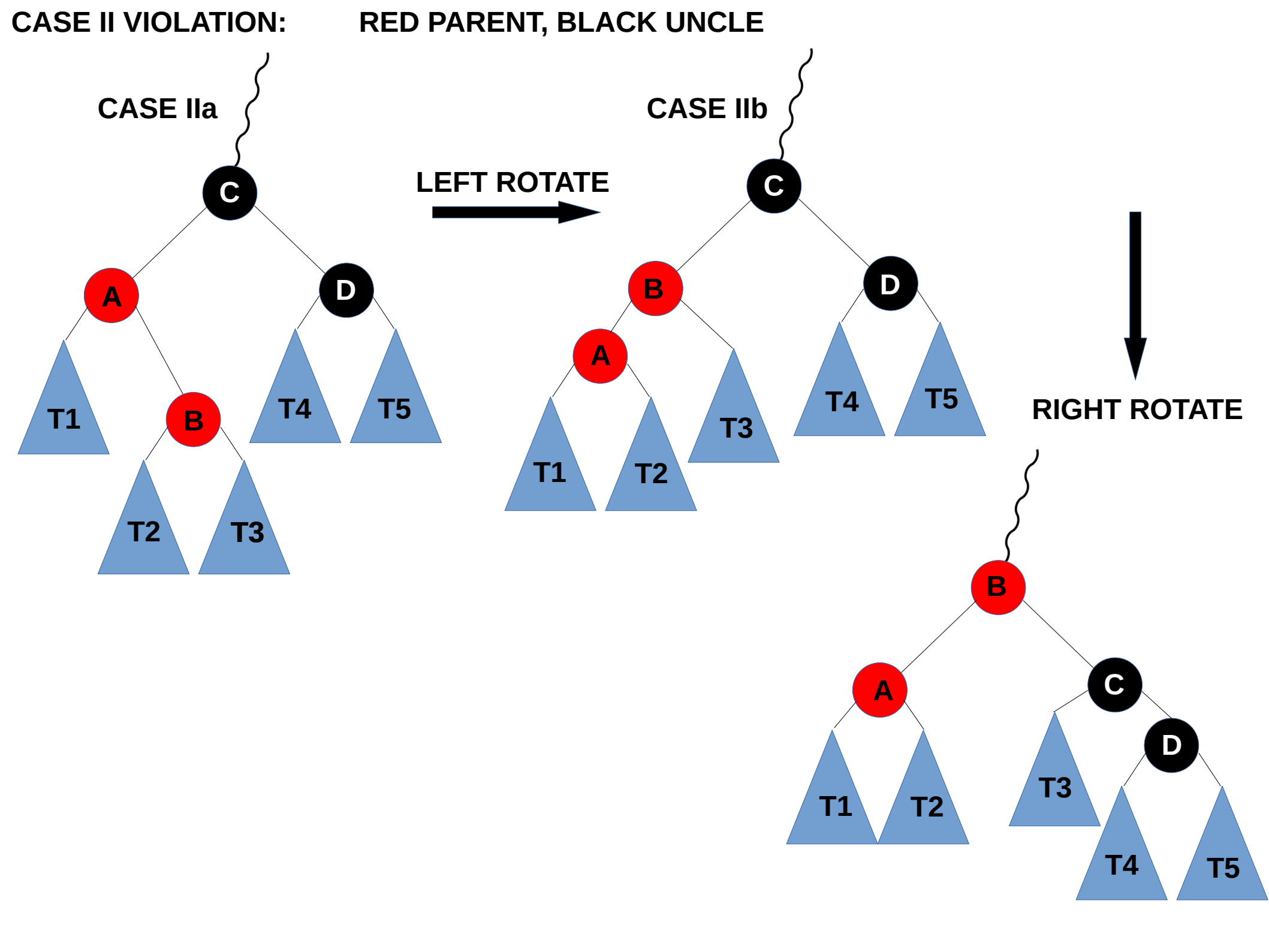


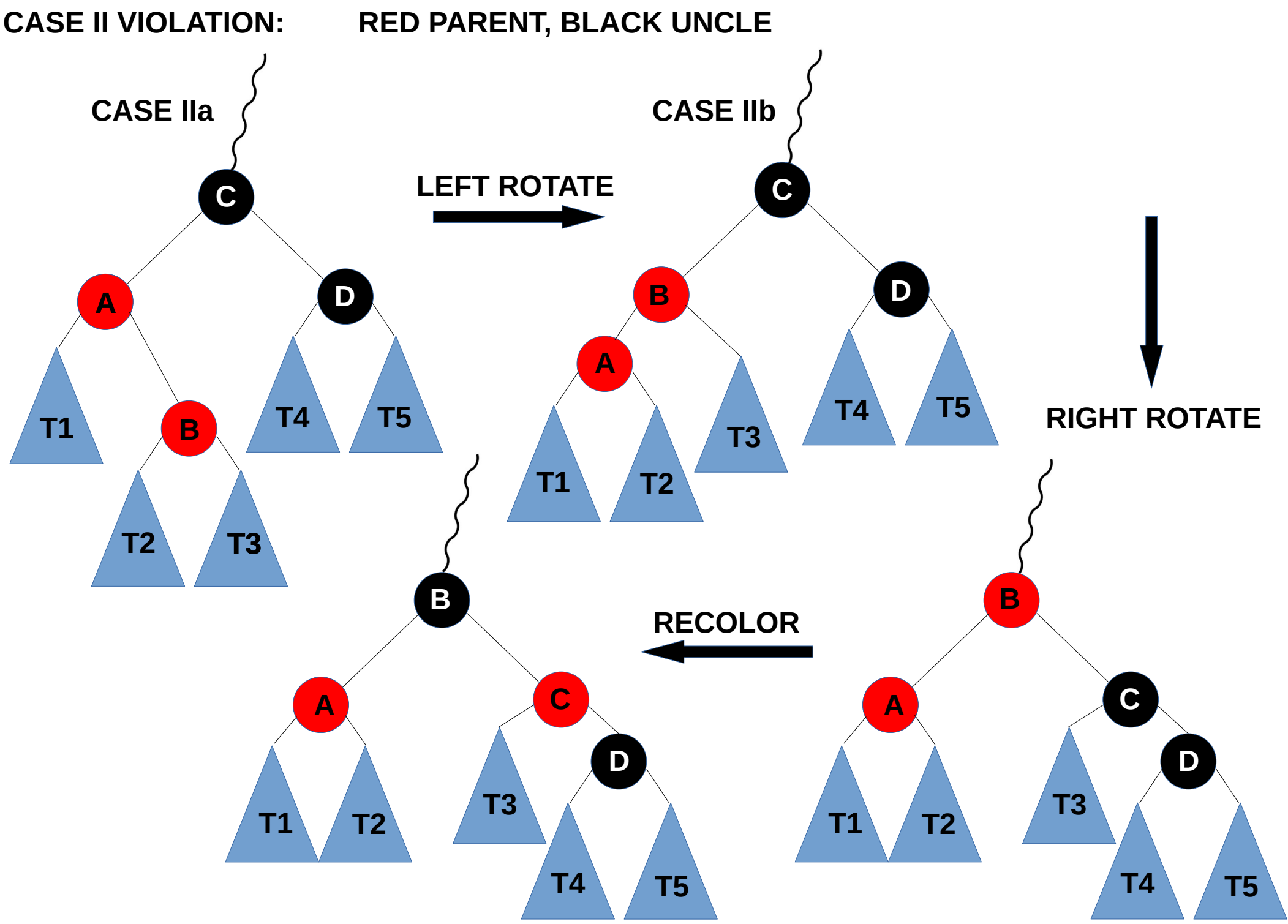
CASE II VIOLATION: RED PARENT, BLACK UNCLE

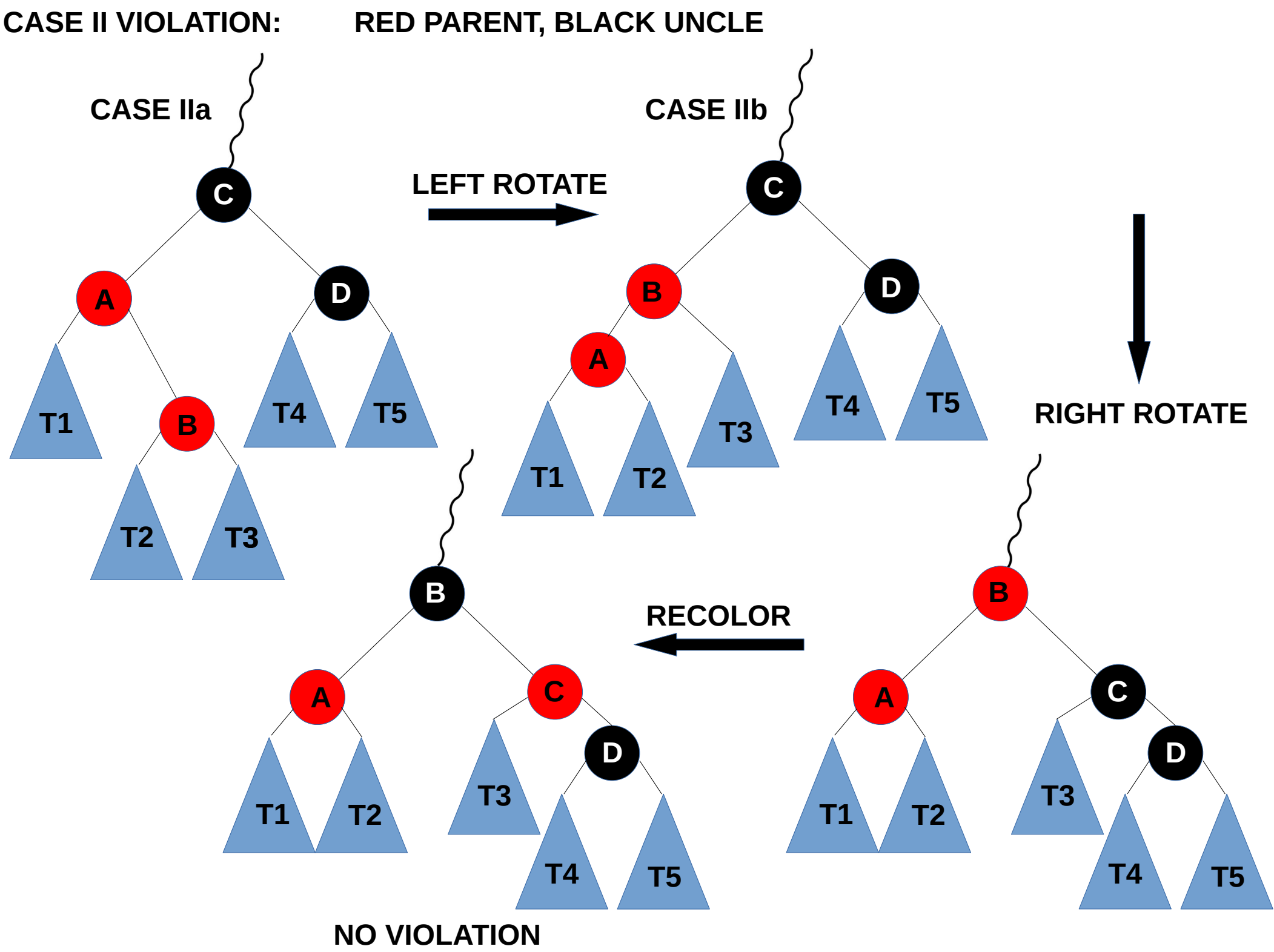


CASE II VIOLATION: RED PARENT, BLACK UNCLE

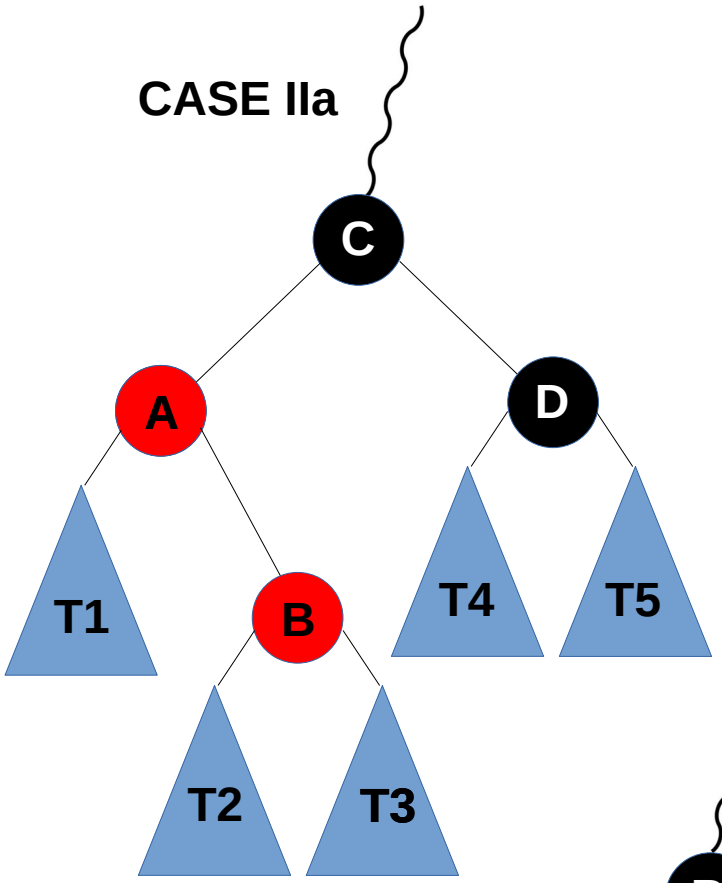



CASE II VIOLATION:
RED PARENT, BLACK UNCLE
CASE IIa
C
A
D
T1
B
T4
T5
T2
T3
LEFT ROTATE
CASE IIb
C
B
D
A
T4
T5
T1
T2
T3
RIGHT ROTATE
B
A
C
T1
T2
T3
D
T4
T5


CASE II VIOLATION:
RED PARENT, BLACK UNCLE
CASE IIa
C
A
D
T1
B
T4
T5
T2
T3
LEFT ROTATE
CASE IIb
C
B
D
A
T4
T5
T1
T2
T3
RIGHT ROTATE
RECOLOR
B
A
C
T1
T2
D
T3
T4
T5
B
A
C
T1
T2
D
T3
T4
T5


CASE II VIOLATION:
RED PARENT, BLACK UNCLE
CASE IIa
CASE IIb
LEFT ROTATE
RIGHT ROTATE
RECOLOR
NO VIOLATION
A
B
C
D
T1
T2
T3
T4
T5

CASE II VIOLATION: RED PARENT, BLACK UNCLE



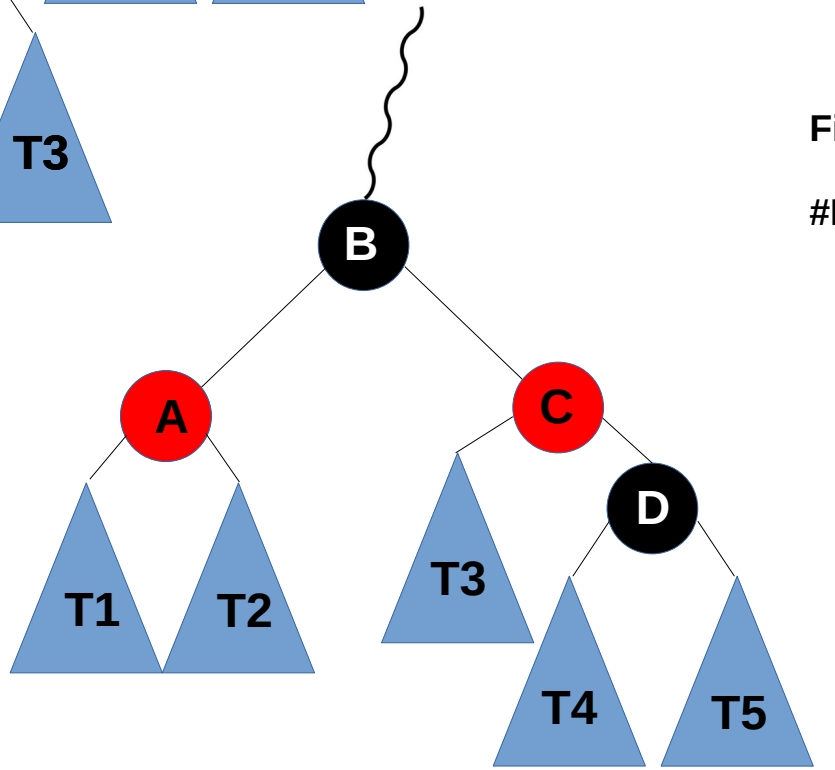
Started with:

$$\#B = \#B^{top} + \#B^{T1}$$
$$= \#B^{top} + \#B^{T2}$$
$$= \#B^{top} + \#B^{T3}$$
$$= \#B^{top} + 1 + \#B^{T4} \quad \text{+1 for D}$$
$$= \#B^{top} + 1 + \#B^{T5} \quad \text{+1 for D}$$

Finished with the same:

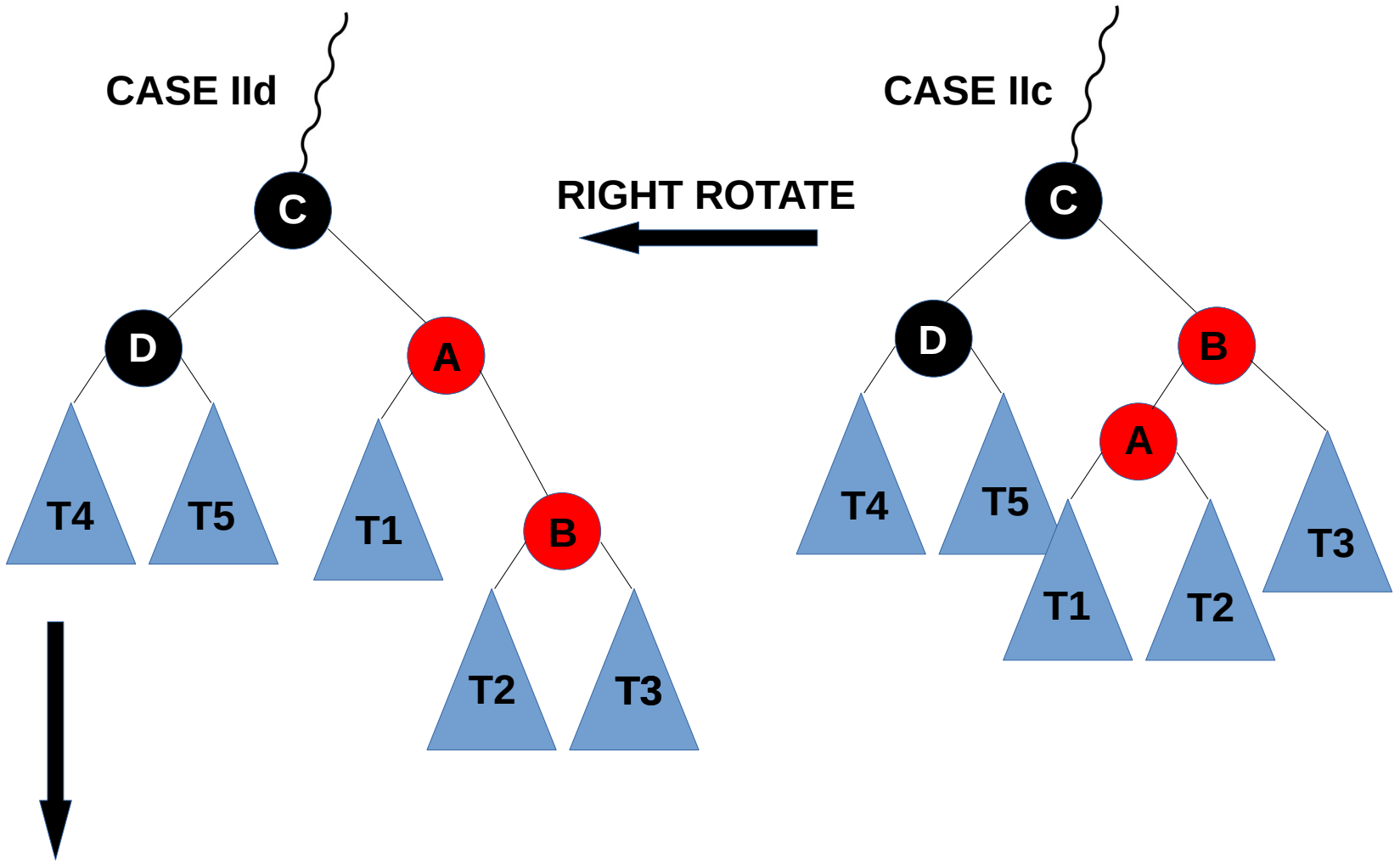
$$\#B = \#B^{top} + \#B^{T1}$$
$$= \#B^{top} + \#B^{T2}$$
$$= \#B^{top} + \#B^{T3}$$
$$= \#B^{top} + 1 + \#B^{T4}$$
$$= \#B^{top} + 1 + \#B^{T5}$$

lost C but gained B
so no change



NO VIOLATION

CASE II VIOLATION: RED PARENT, BLACK UNCLE



LEFT ROTATE

and so on...