

B3R-Tree

→ Node Structure

1. Representation:

PARENT NODE		
K_1		K_2
LEFT CHILD	CENTER CHILD	RIGHT CHILD

Attributes

Pointers → L, C, R Child Nodes
 Keys ← K_1 & K_2
 leaf

2. Insert Keys:

	null null	K_1 null	null Key
1	$K[0] \leftarrow K_1$	$K[1] ? \text{null} ?$	$K[0] ? \text{null}$
2		$K_1 < K_2 ?$	$K_1 > K_2$
3		$K_1 \ K_2 \mid K_2 \ K_1$	$K_1 \ K_2 \mid K_2 \ K_1$

Insert (Value)
 if (noSpace) → exit

if (K_1 empty || $K_1 > \text{Value}$)
 if ($K_1 \neq \text{null}$)
 $K_2 \leftarrow K_1$
 $K_1 \leftarrow \text{Value}$
 else
 $K_2 \leftarrow \text{Value}$

3. Remove

remove Key (value)

if ($K_1 \neq \text{null}$ & $K_1 = V$)

$K_1 \leftarrow \text{null}$

if ($K_2 \neq \text{null}$)

$K_1 \leftarrow K_2$

$K_2 \leftarrow \text{null}$

else if ($K_2 \neq \text{null}$ & $K_2 = V$)

$K_2 \leftarrow \text{null}$

IF KEY
NEEDED

remove Key (value)

if ($K_1 \neq \text{null}$ & $K_1 = V$)

pop ← K_1

$K_1 \leftarrow \text{null}$

if ($K_2 \neq \text{null}$)

$K_1 \leftarrow K_2$

$K_2 \leftarrow \text{null}$

return pop

else if ($K_2 \neq \text{null}$ & $K_2 = V$)

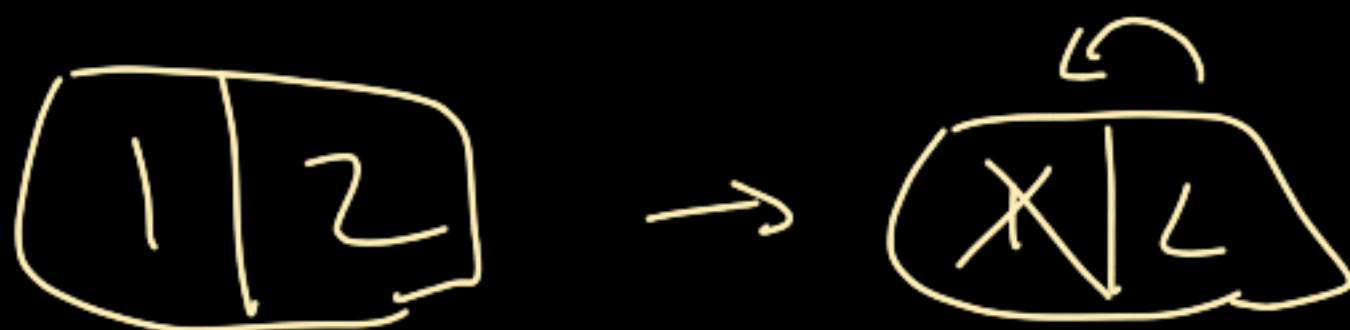
pop ← K_2

$K_2 \leftarrow \text{null}$

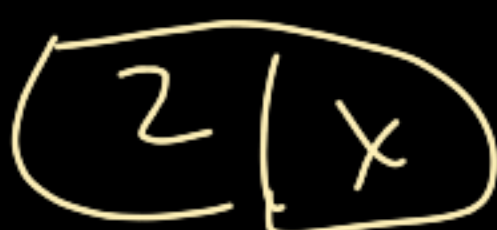
return pop

else

return null



↓



→

