

Indexing notes

Index on
 $R(B)$



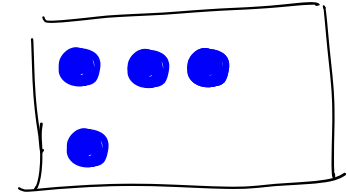
Select B
From R
ORDER BY B



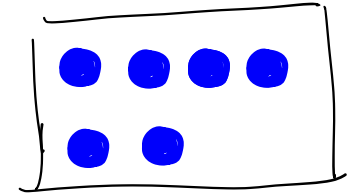
B value →
pointer
to tuple

Relation $R(A,B,C,D)$

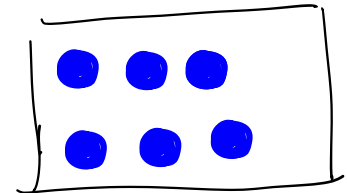
page 1



page 2

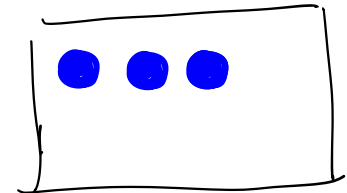


page 3

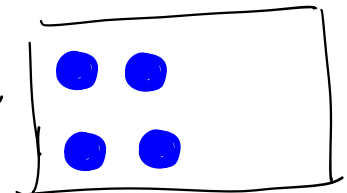


⋮

page $K-1$



page K



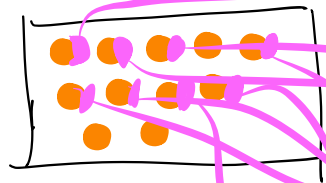
select *
FROM R
WHERE
B=5;

<assume
B is unique>

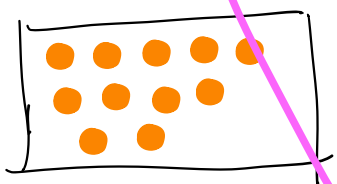
Index on R(B)

Relation R(A,B,C,D)

index
page 1

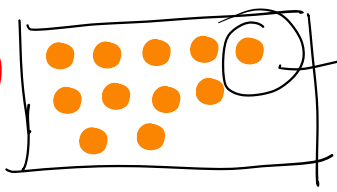


index
page 2



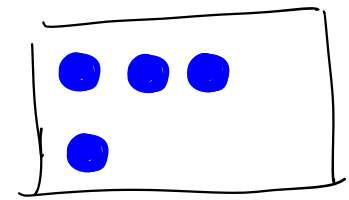
...

index
page (M)

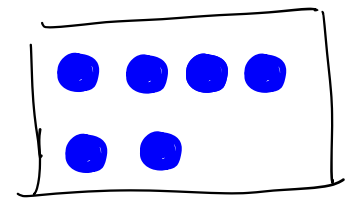


M << K

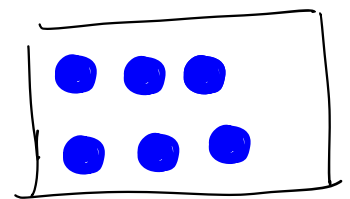
page 1



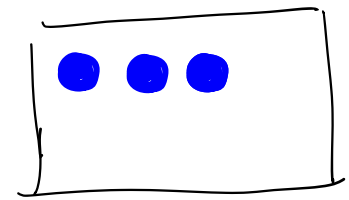
page 2



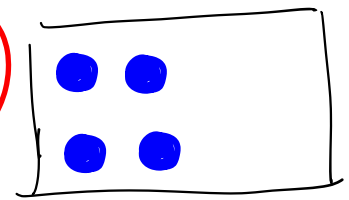
page 3



page K-1

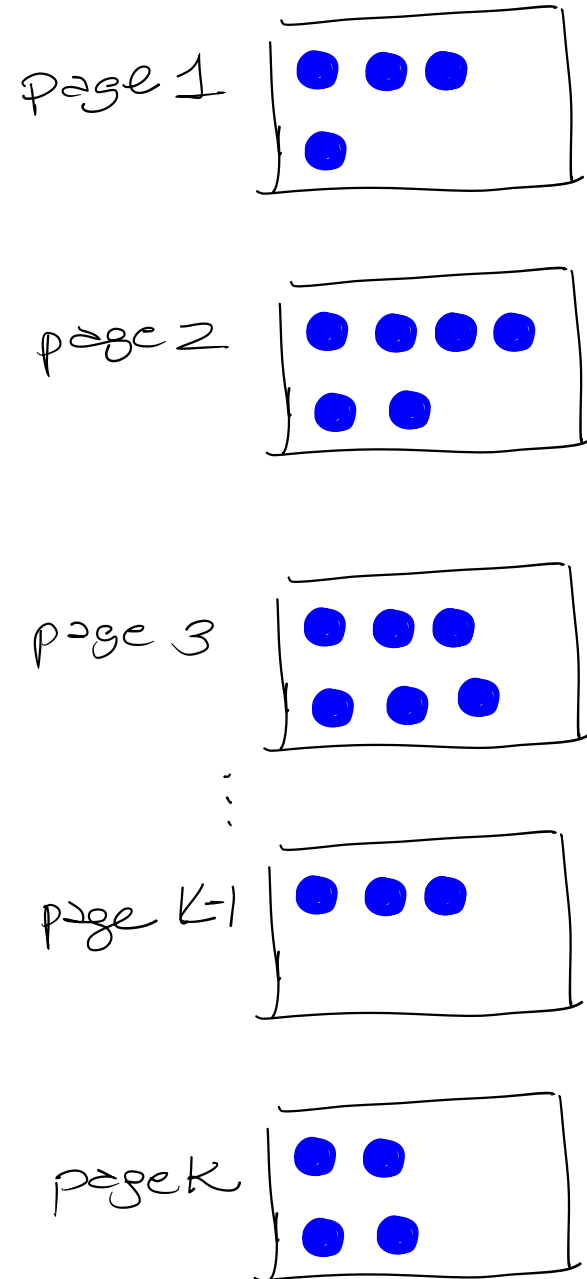
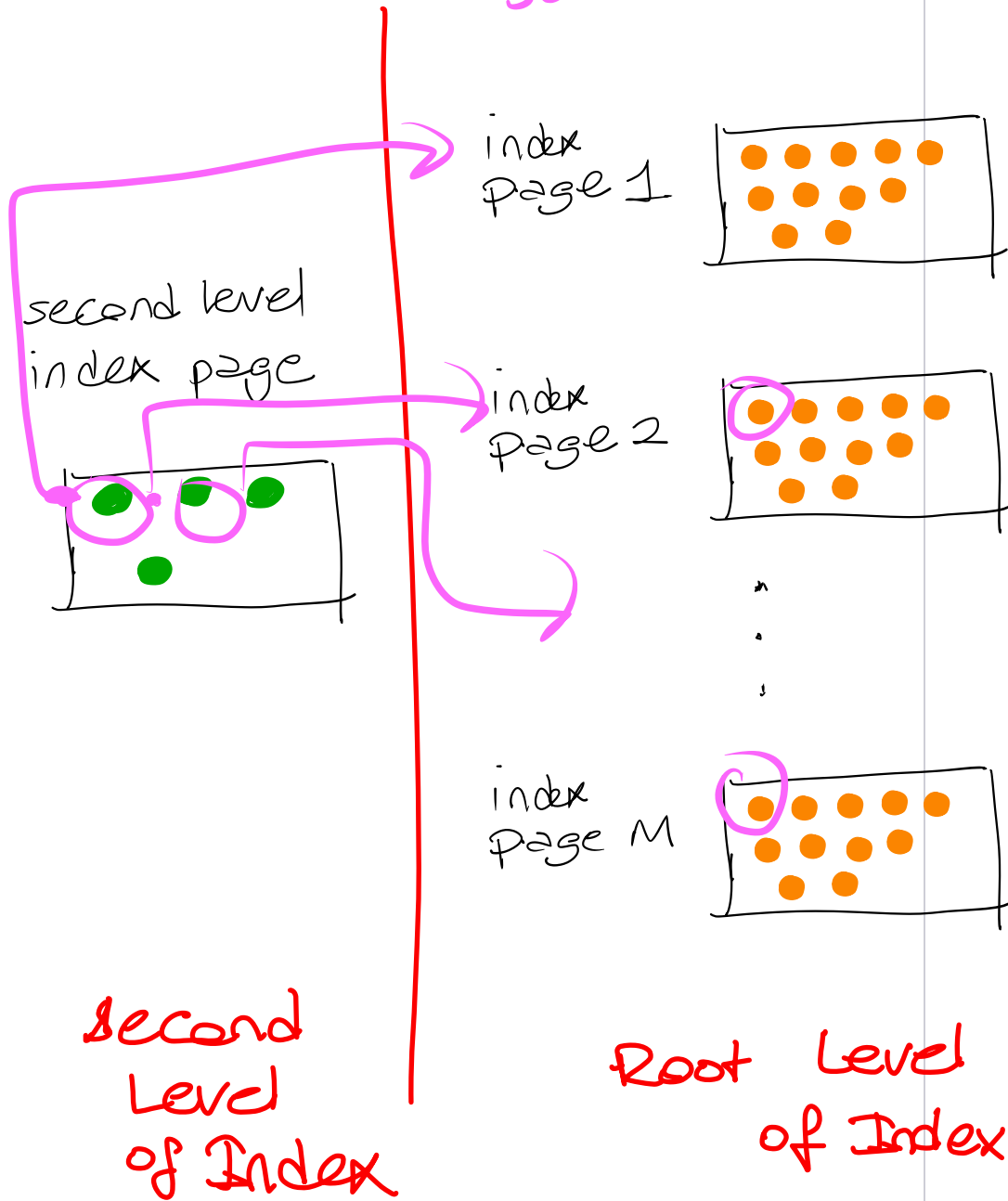


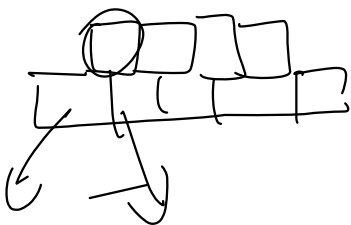
page (K)



Index on $R(B)$
sorted B

Relation $R(A, B, C, D)$

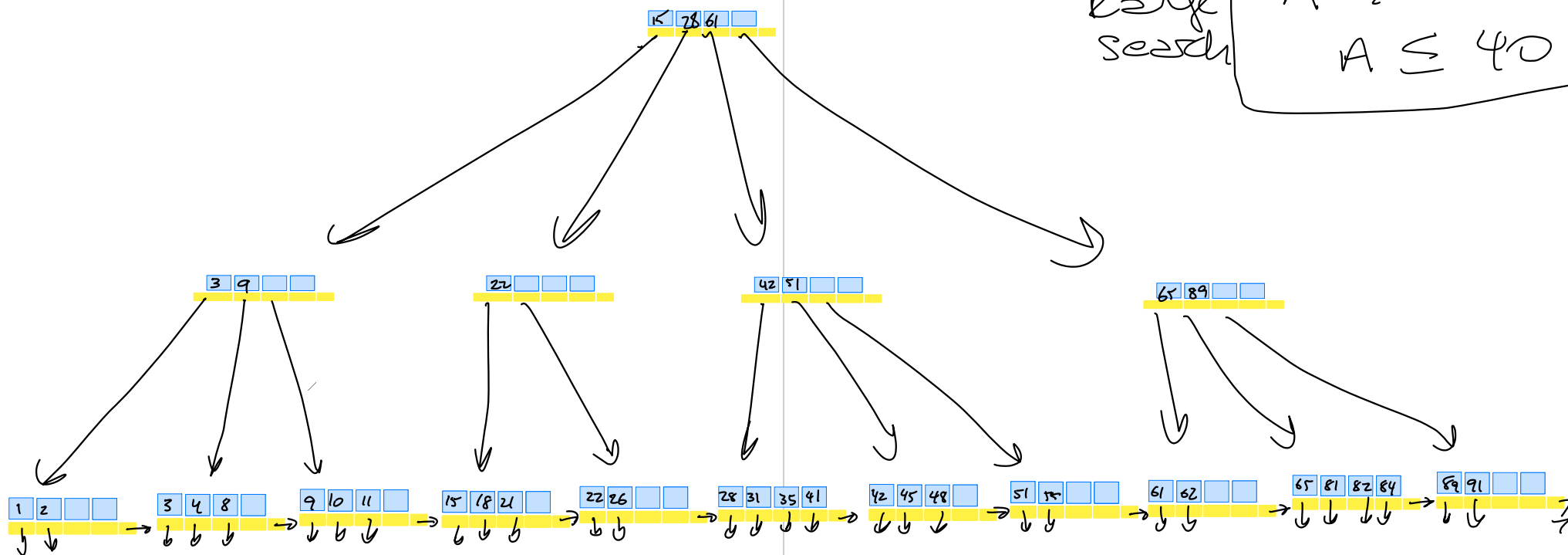


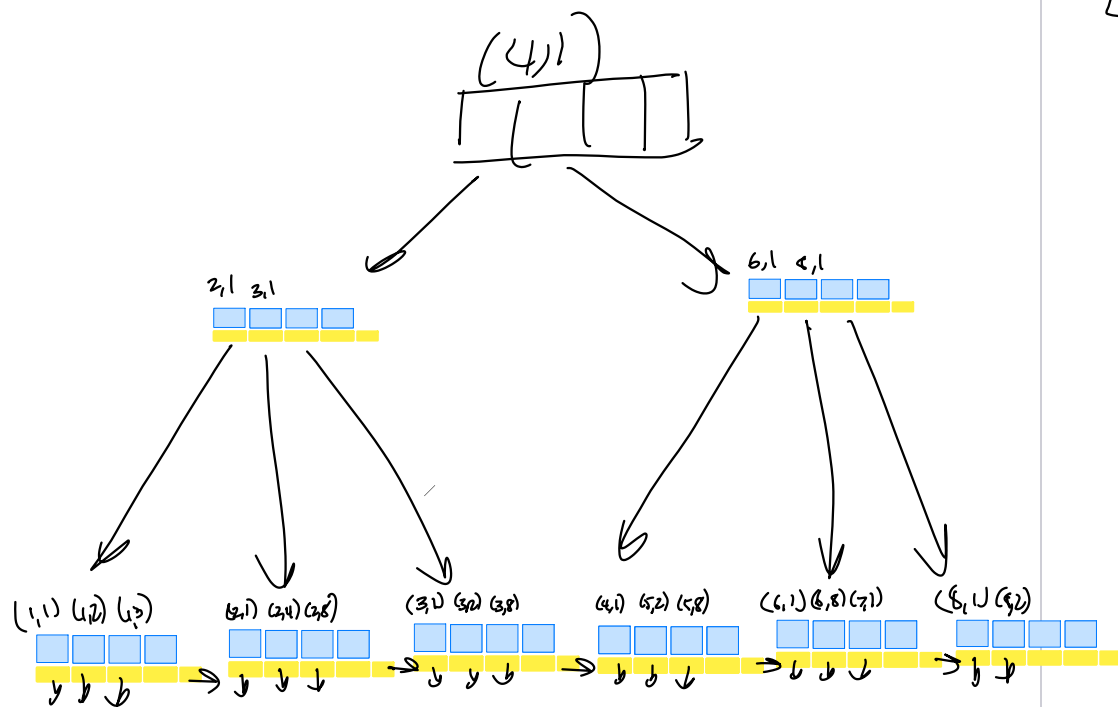


Index on $R(A)$
int

Range search

$A = 10$
$A \geq 10 \ \&$
$A \leq 40$





Index on $R(A,B)$

$A=2$ and $B=4$

$A=2$

$A \geq 2$ and $A \leq 4$

$A \geq 2$ and $A \leq 4$
and $B=4$

→ scan
between

$B=4$

$(2,4) -$
 $(4,4)$

Insert 16

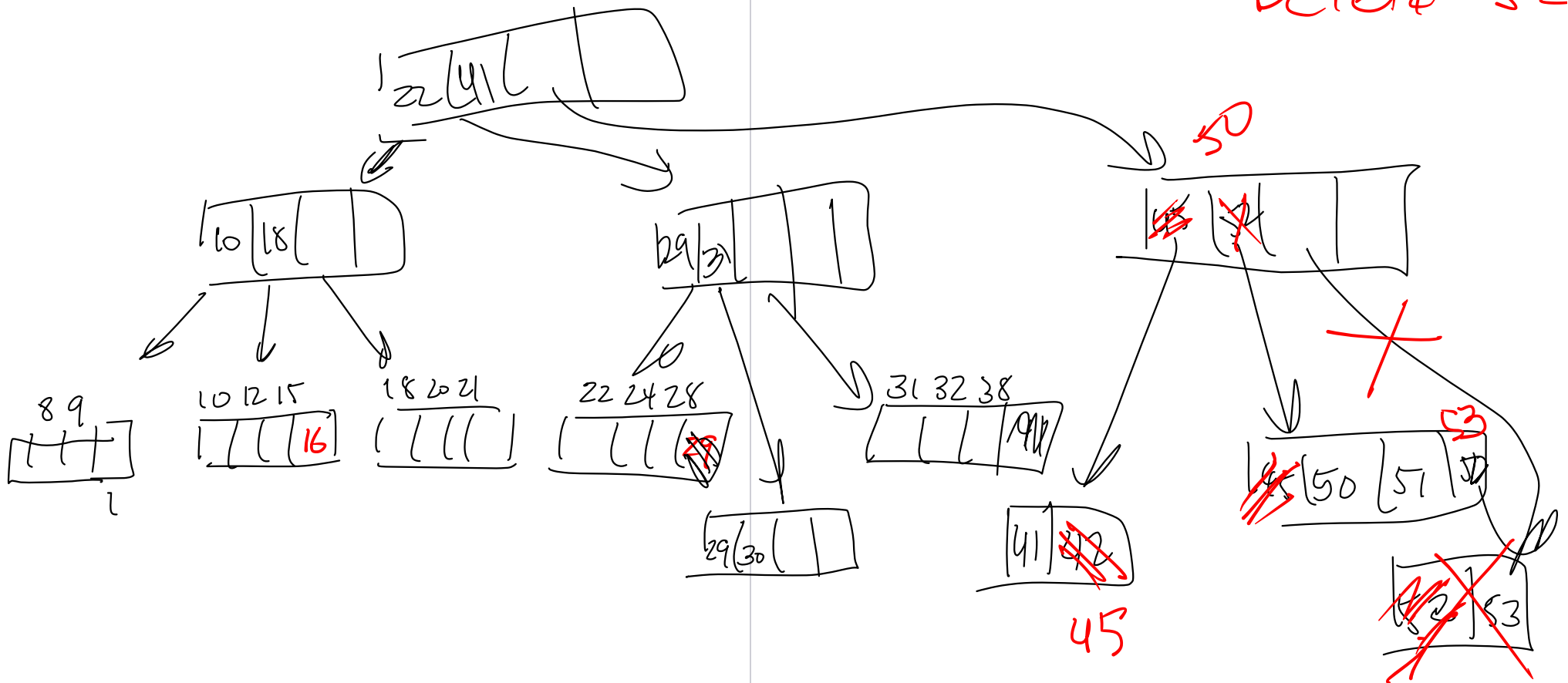
Insert 29

Insert 30

Delete

42

Delete 52



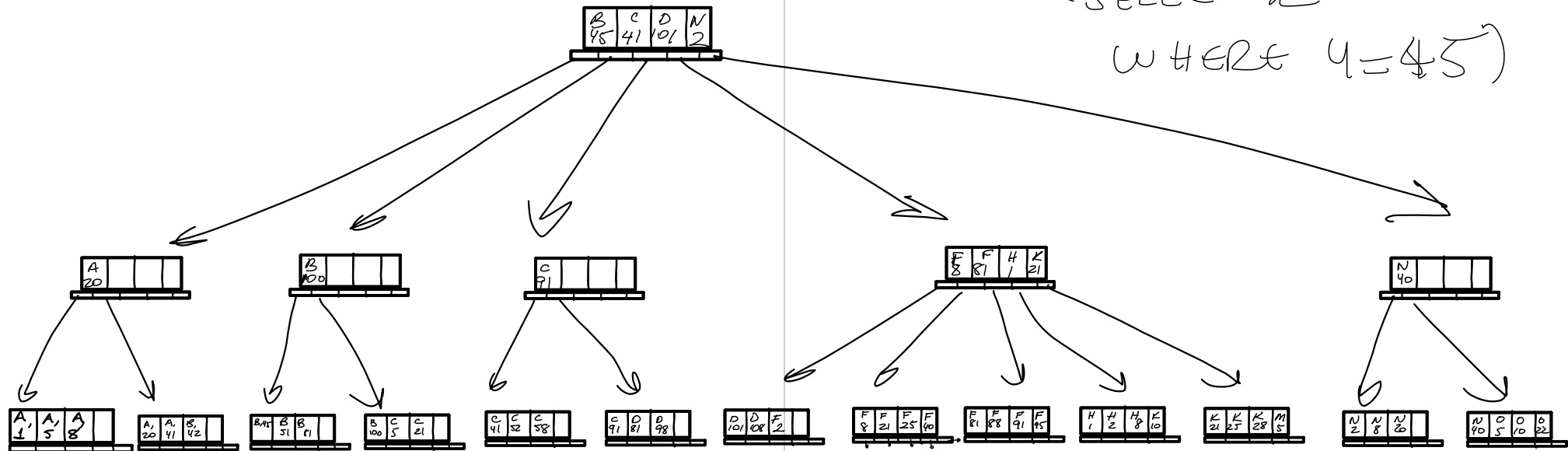
$n=4$ leaf 4 tuples (key value + pointer to tuple)
 1 sibling index node
 internal 4 key values
 5 index nodes

Index on $R(x, y)$

Half full:

leaf \rightarrow 2 tuples min (4 tuples max)

Internal \rightarrow 2 nodes indexed min (5 max)



Search:

$X=A$

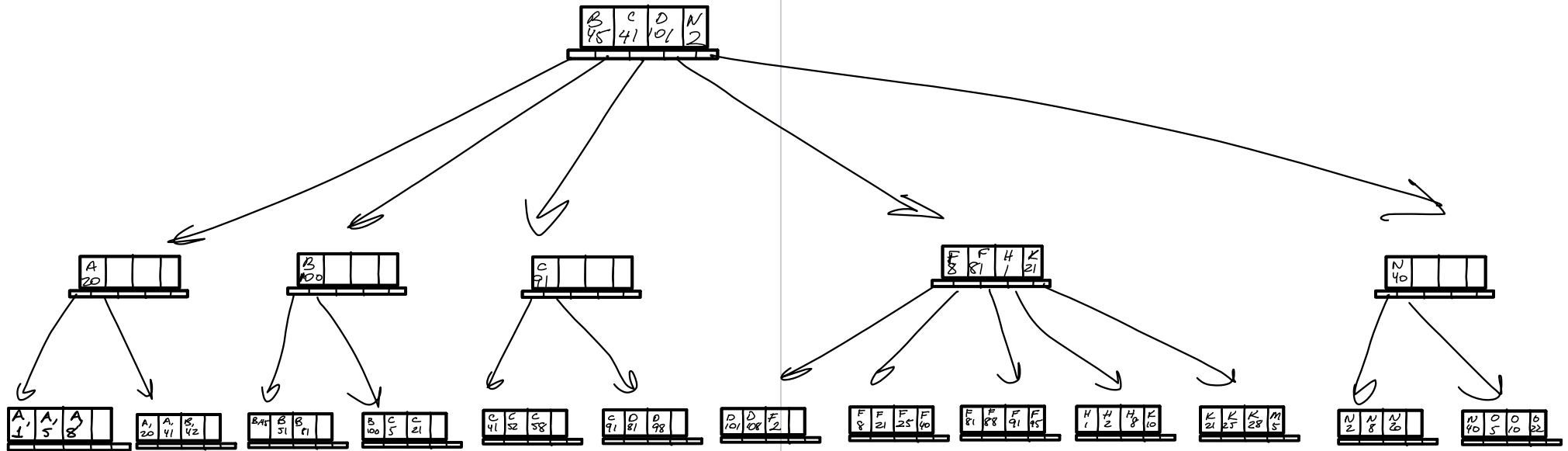
(select B FROM R WHERE $X=A$)

$y=45$

(select x FROM R WHERE $y=45$)

$y \geq 45$

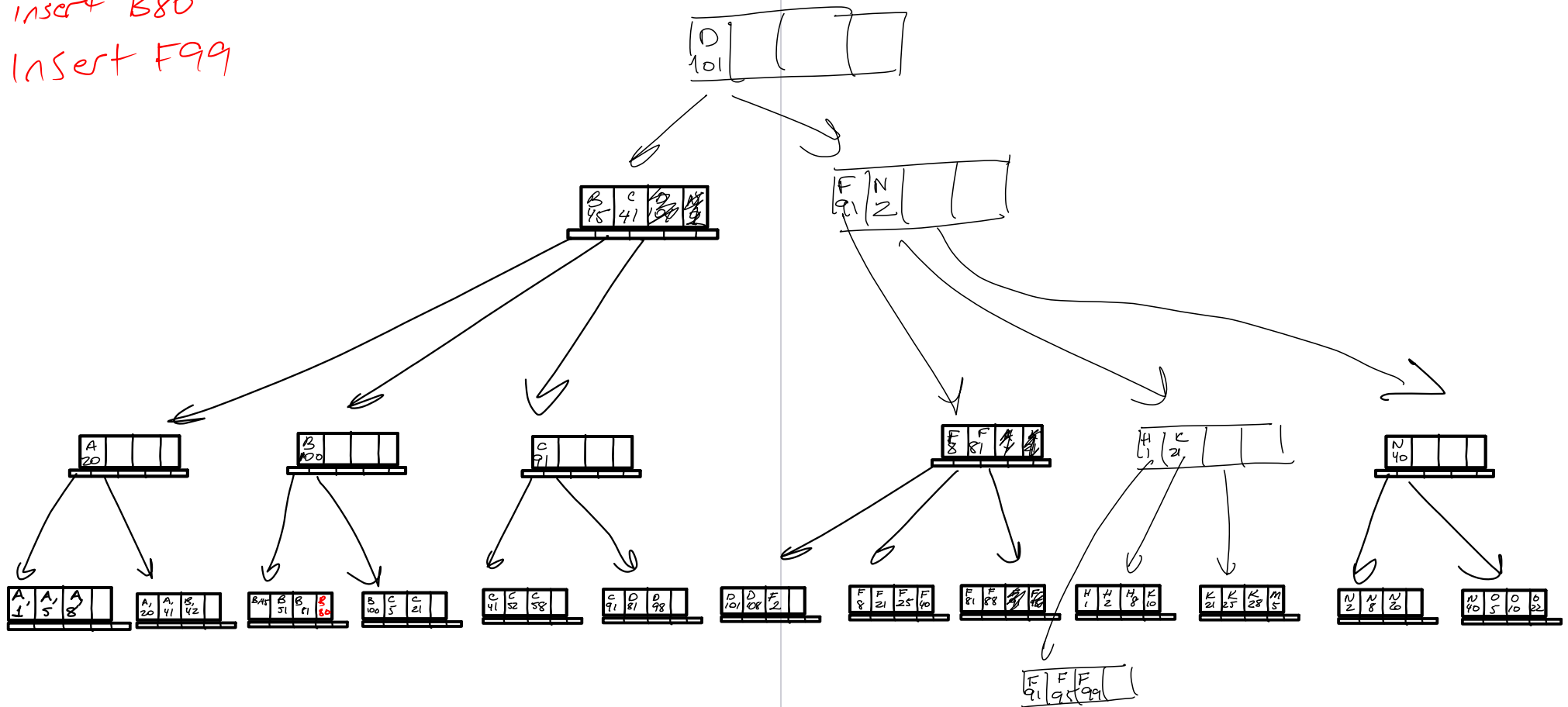
Insertion Example 1



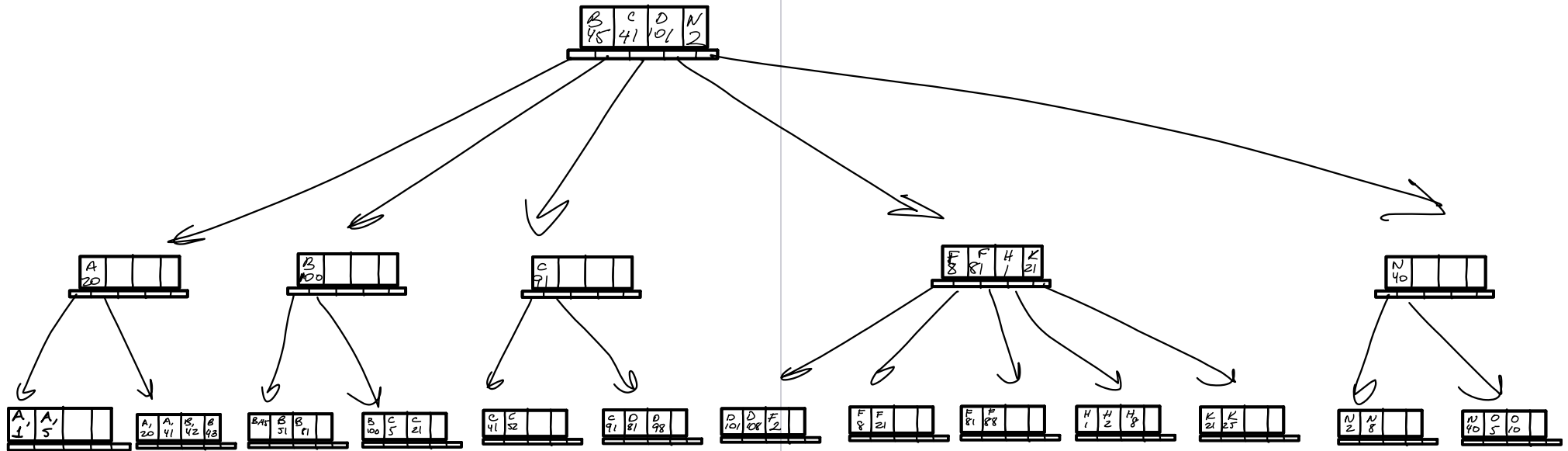
Insertion Example 1

Insert B80

Insert F99



Deletion Example 1



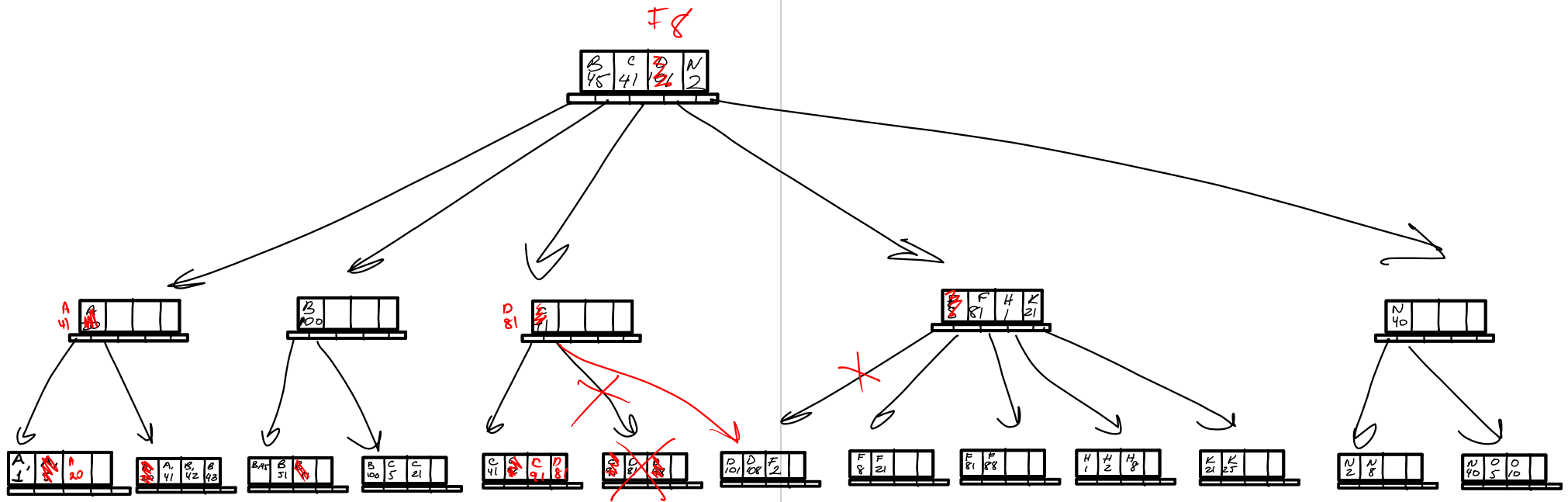
Deletion Example 1

Delete B81

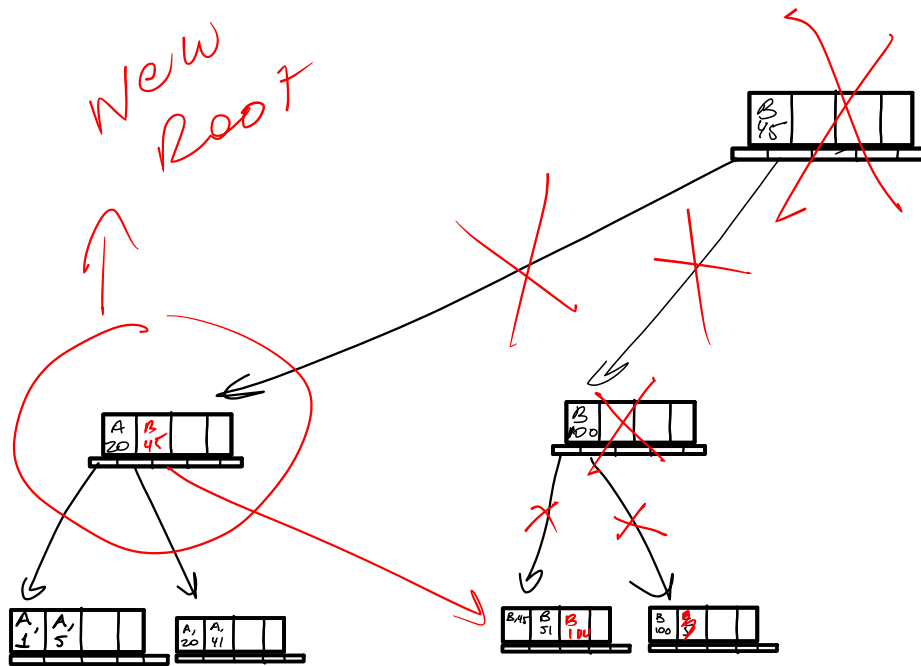
Delete A5

Delete C52

Delete D98

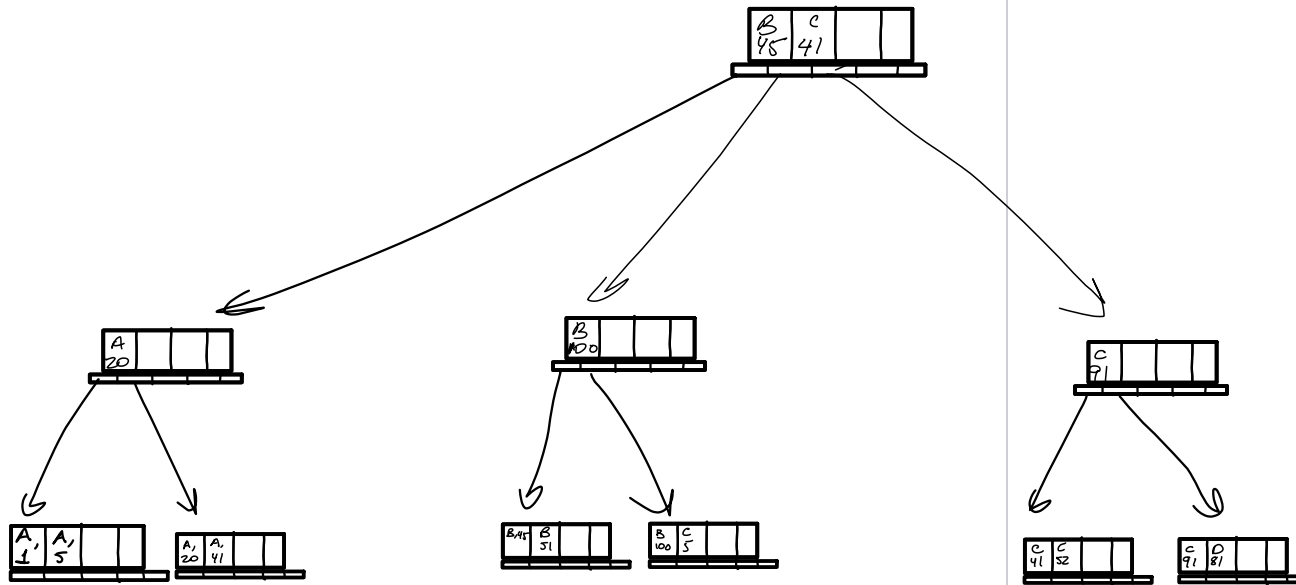


Deletion Example 2

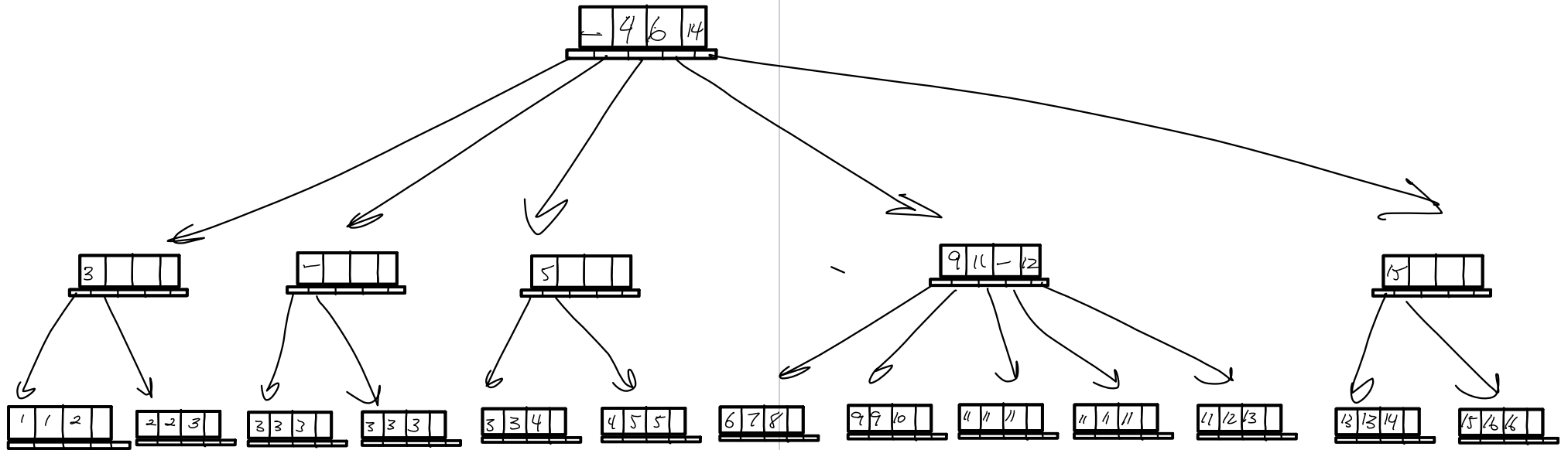


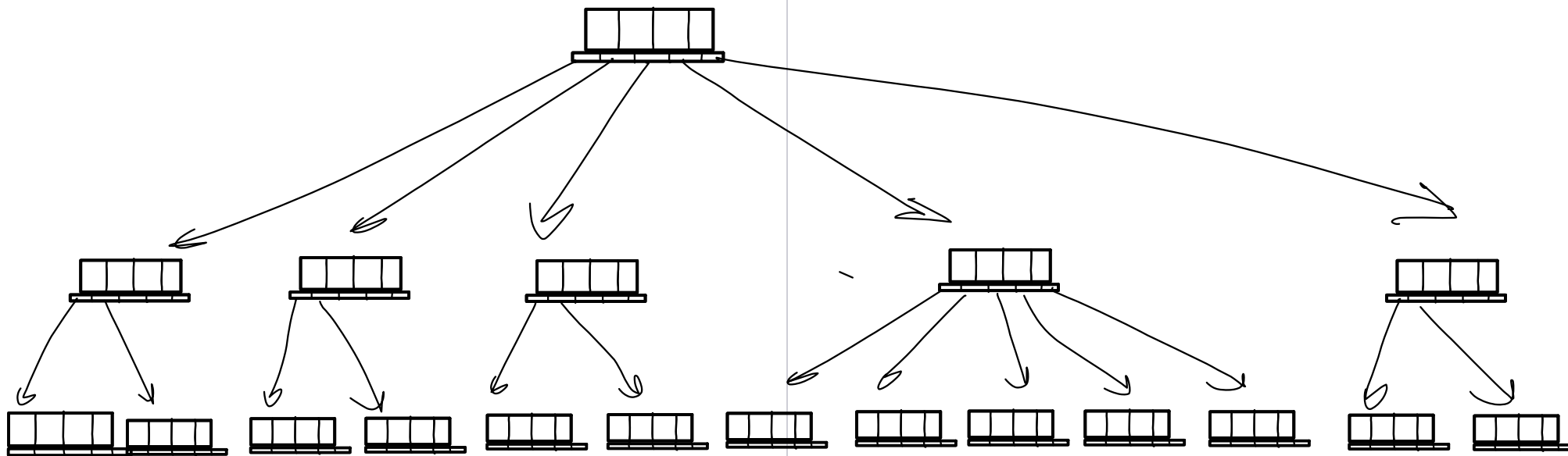
Delete c5

Deletion Example 2



Indexing duplicate values





$(4, x)$

$P3, P5$

$P4$

$P1$

$P6$

$P2$

$P7$

$P3$

$P4$

$P5$

$P1$

$P2$

$P6$

$P7$

$P1$

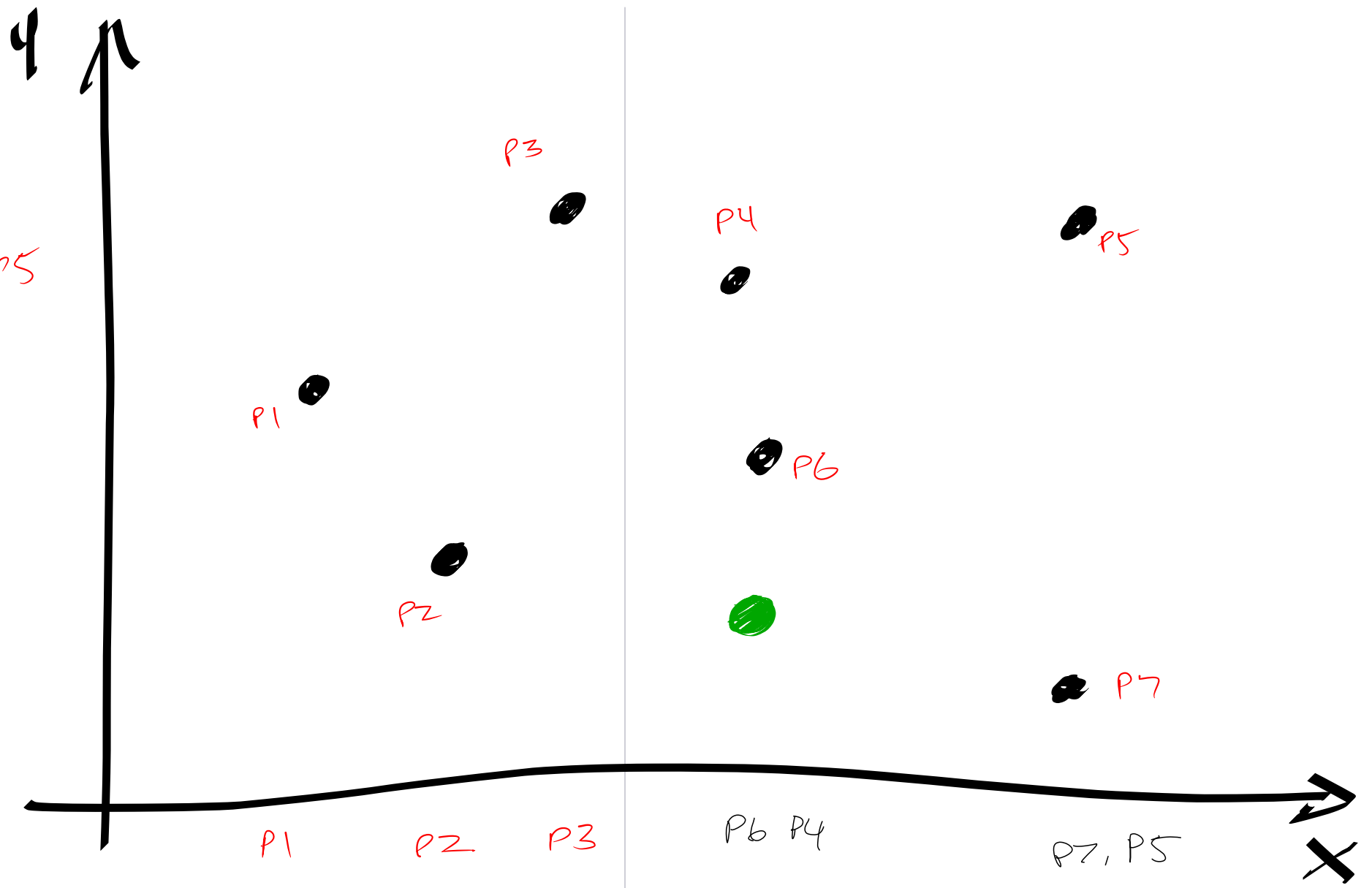
$P2$

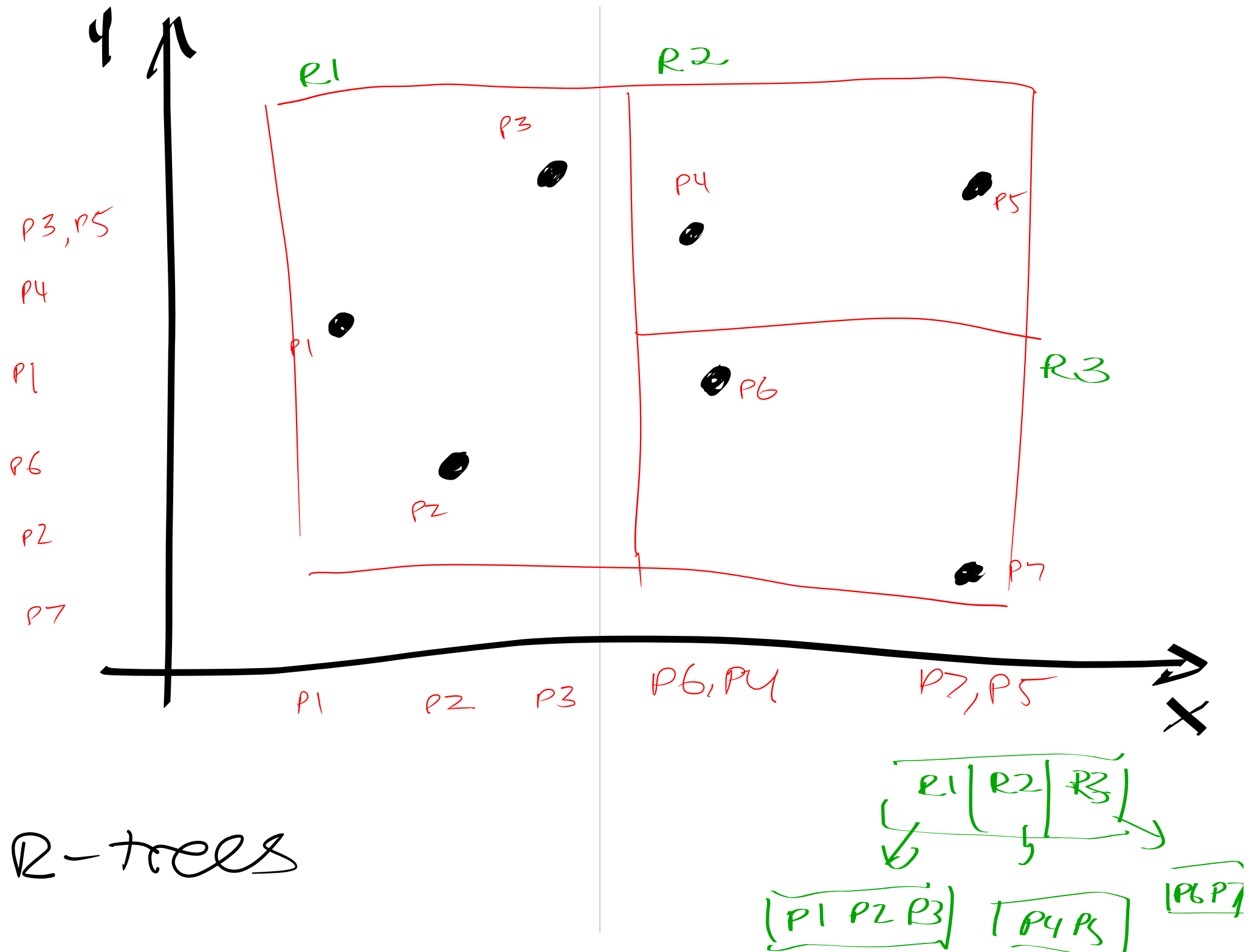
$P3$

$P6, P4$

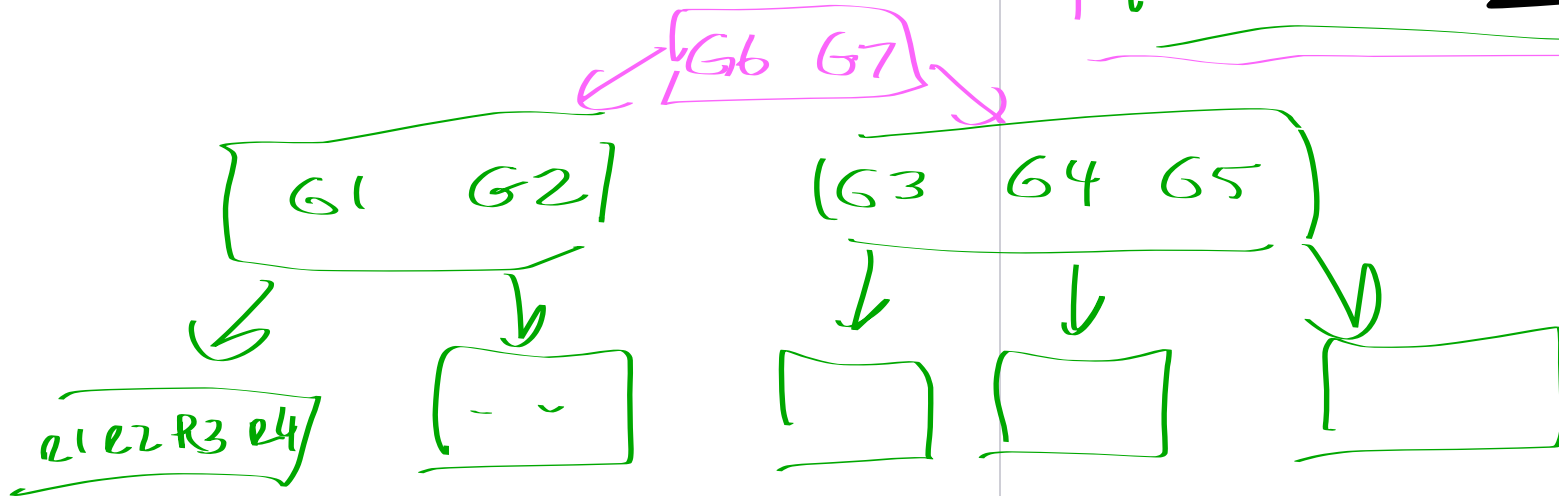
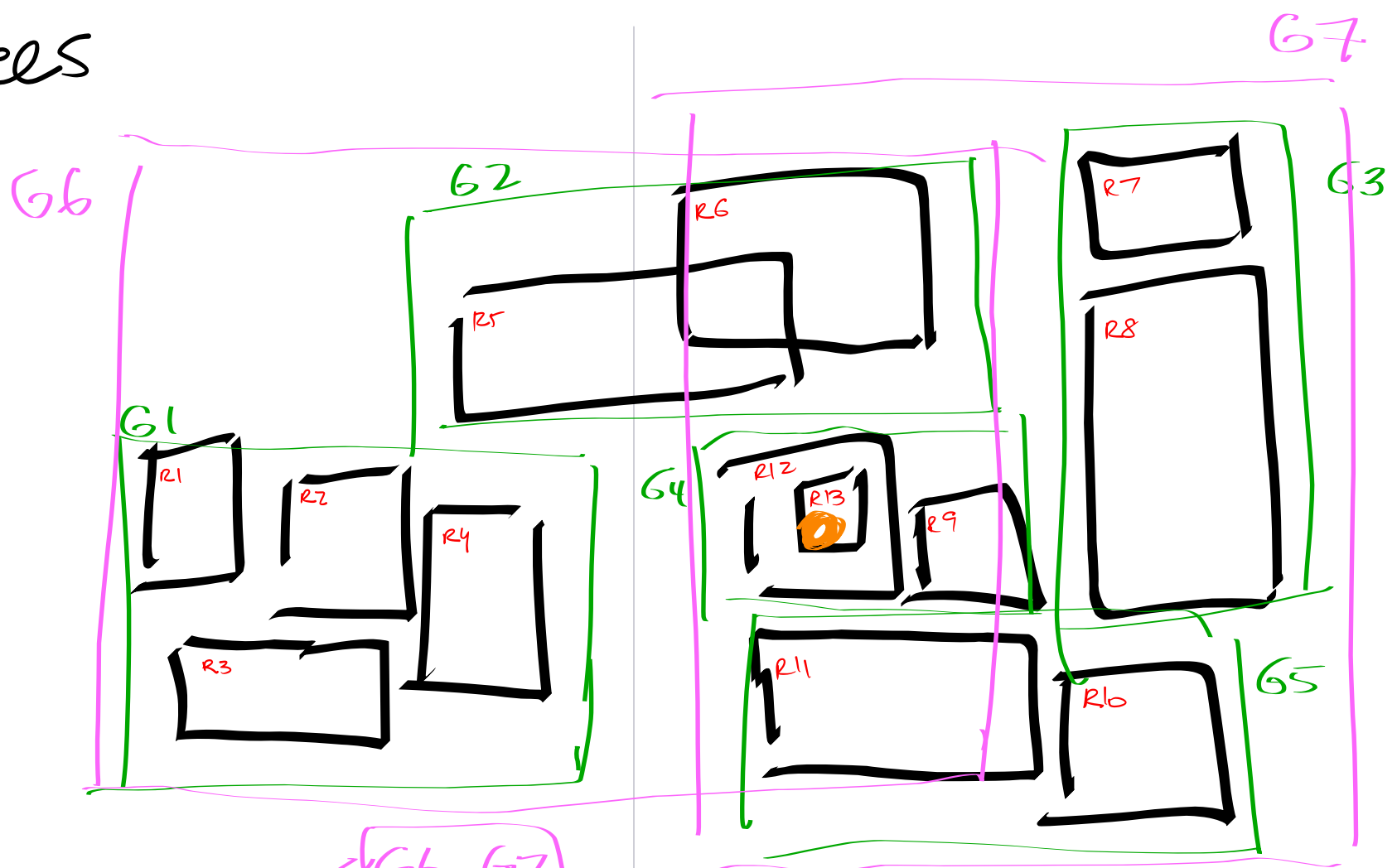
$P7, P5$

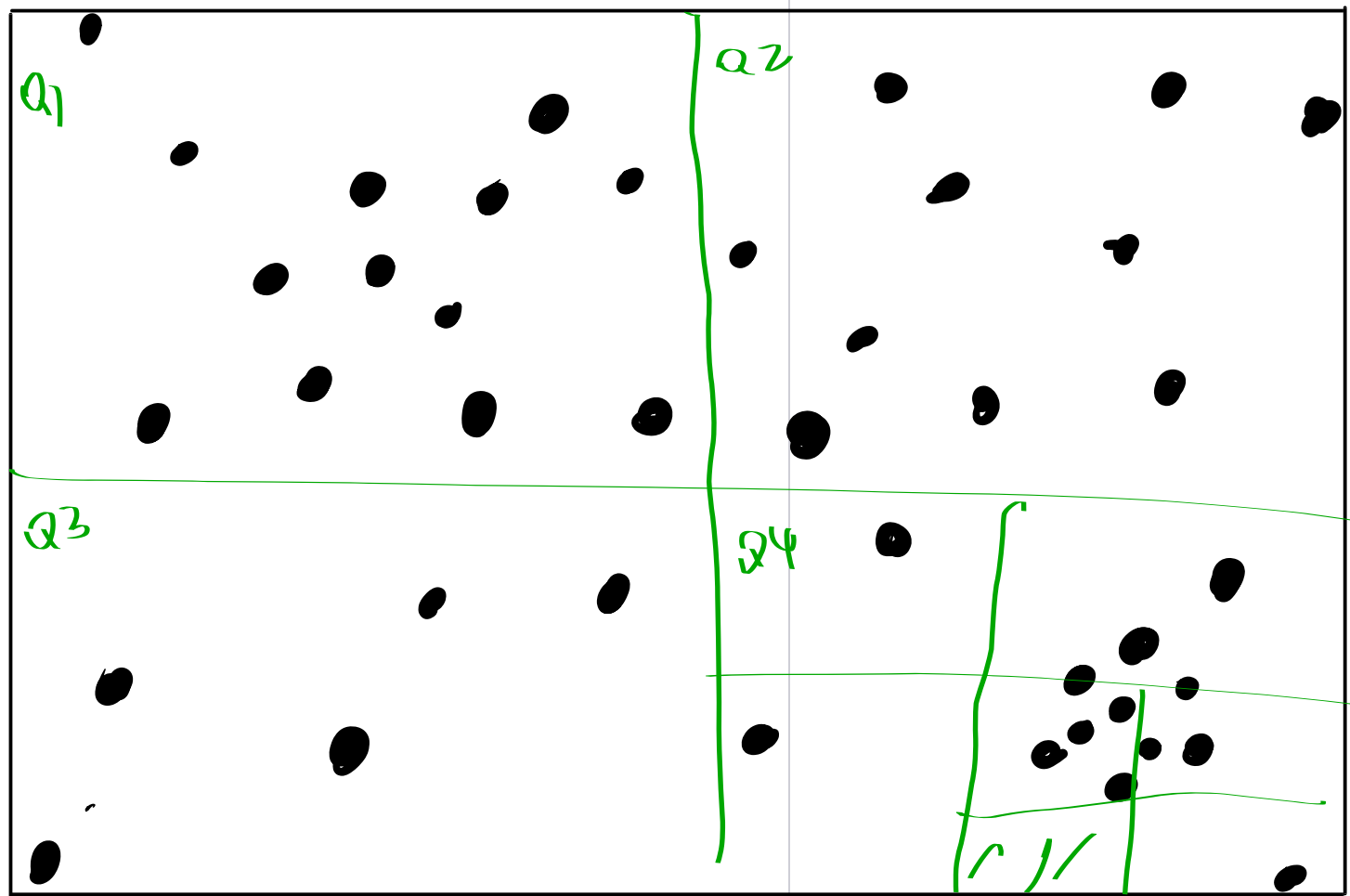
$(x, 4)$



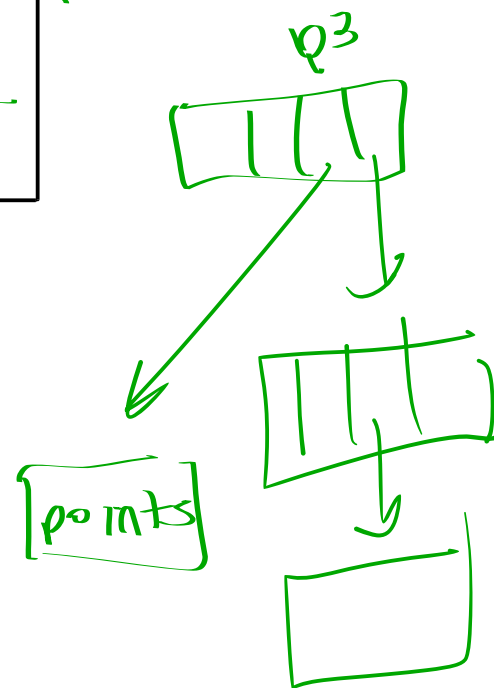


R-trees





QUADTREES
(unbalanced)



Hashing

$$t \rightarrow h(t, A) \rightarrow h_1$$

$$h(t) \rightarrow$$

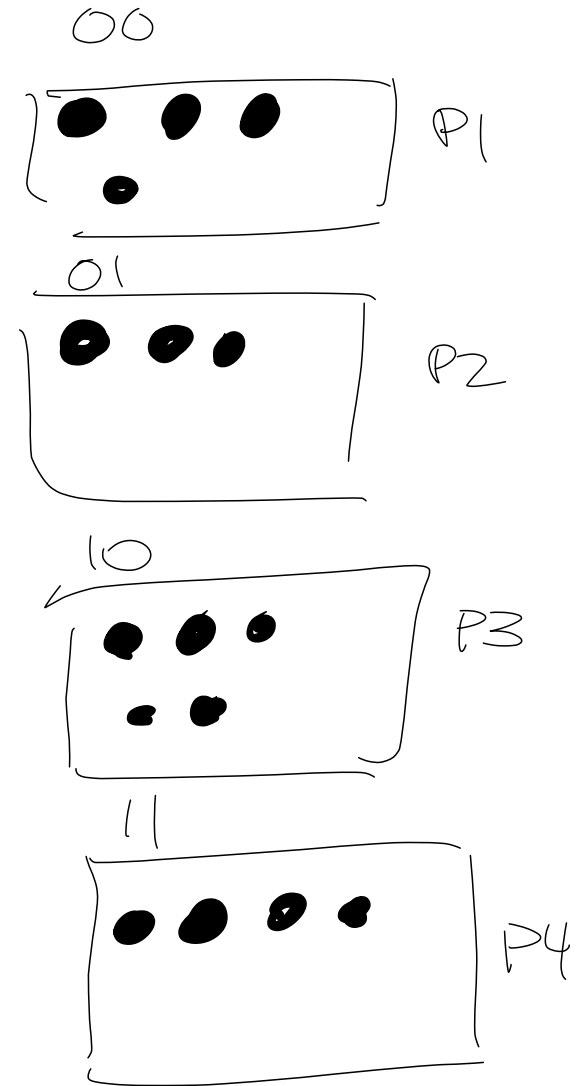
Search

SELECT *

FROM R

WHERE $A = 25$

$$h(25) \rightarrow \dots \rightarrow 10$$



Hashing

