EXERCISE 1

Construct a B+-tree for the following set of values

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

Assume that the tree is initially empty and values are inserted in ascending order.

1) Construct B+-trees for the cases where the number *m* of pointers that will fit a node is as follows:

   a. Four

   b. Seven

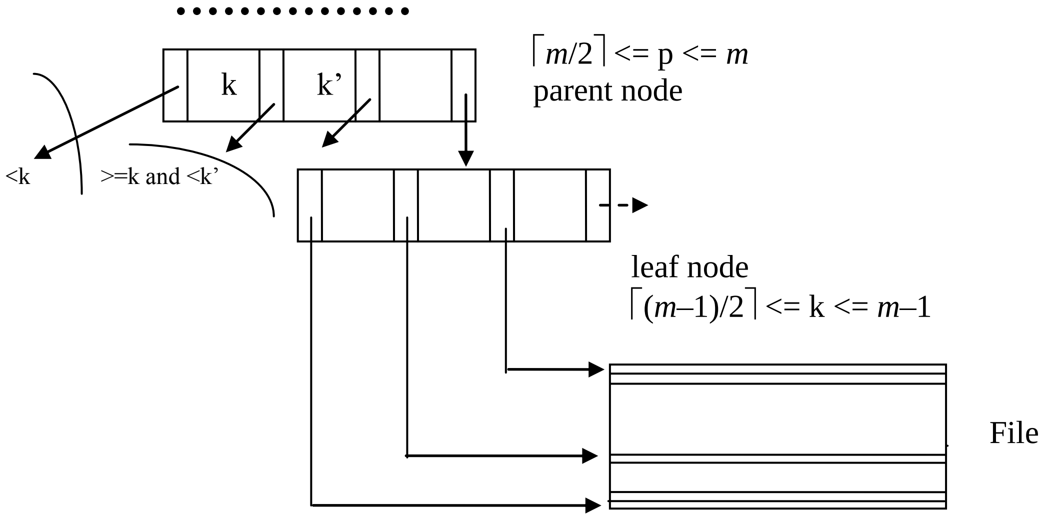
2) Shows the form of the B+-tree after each operation of the sequence:

   Insert 9; Insert 10; Insert 8; Insert 6; Insert 1; Insert 4

   for the case *m*=4.

Solution

Point 1)



B+-tree:

Not a root or a leaf node: $\lceil m/2 \rceil <= p <= m$, where p is a pointer.

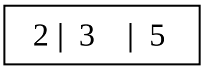
Leaf node: $\lceil (m-1)/2 \rceil <= k <= m-1$, where k is a key.

Root: p>= 2   if it is not a leaf

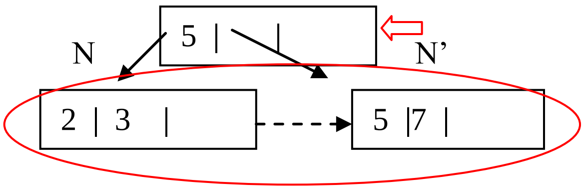
  0<=k<= (*m*–1) if it is a leaf.

a) *m*=4	Internal node: 2<=p<=4	Root: p>= 2 if it is not a leaf;
	Leaf node: 2<=k<=3	0<=k<= 3 if it is a leaf

B+tree: Insert 2, 3, 5



B+tree: Insert 7 (leaf split)

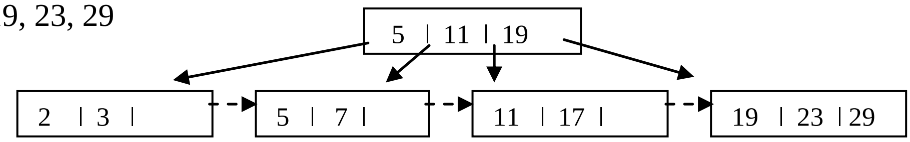


Split a leaf node:

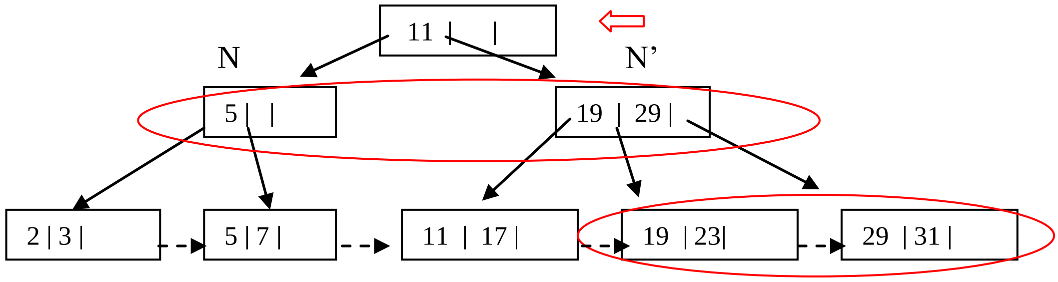
Let $K_1, \ldots, K_m$ be the set of keys in the ascending order (i.e. 2, 3, 5, 7)

- Node N: $K_1, \ldots, K_{\lceil m/2 \rceil -1}, K_{\lceil m/2 \rceil}$ (i.e. 2, 3)
- Node N': $K_{\lceil m/2 \rceil +1}, \ldots, K_m$ (i.e. 5, 7)
- Insert $K_{\lceil m/2 \rceil +1}$ into the parent node (i.e. 5)

B+tree: Insert 11, 17, 19, 23, 29



B+tree: Insert 31 (leaf node split + non leaf node split)



Split an internal node N:

Let $K_1, \ldots, K_m$ be the set of keys in the ascending order (i.e. 5, 11, 19, 29)

- Node N: $K_1, \ldots, K_{\lceil m/2 \rceil -1}$ (i.e. 5)
- Node N': $K_{\lceil m/2 \rceil +1}, \ldots, K_m$ (i.e. 19, 29)
- Insert $K_{\lceil m/2 \rceil}$ into the parent node of N (i.e. 11)

Note that since values are inserted in ascending order, leaves have the minimum number of keys except the last leaf.

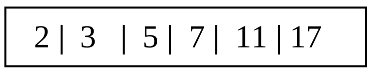
b) Seven.

m=7 Intermediate node : 4<=p<=7 Leaf node: 3<=k<=6

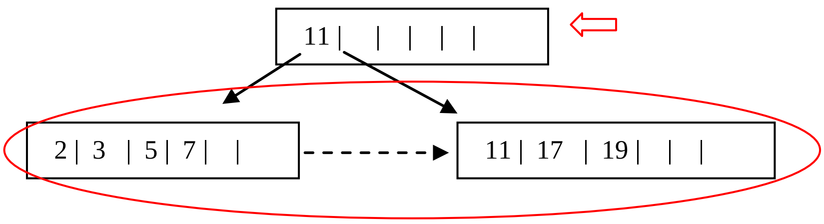
Root: p>= 2 if it is not a leaf ;

0<=k<= 6 if it is a leaf

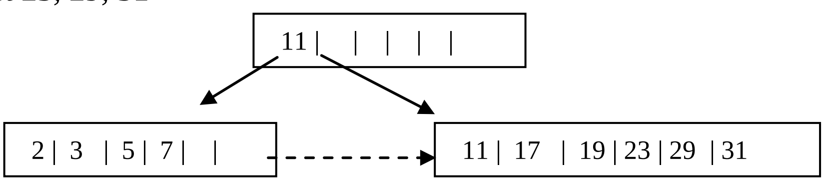
B+tree: Insert 2, 3, 5, 7, 11, 17



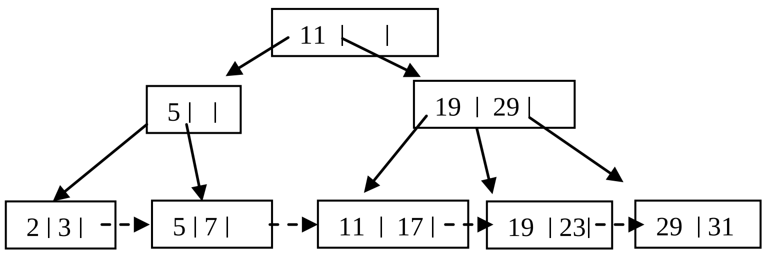
B+tree: Insert 19 (leaf split)



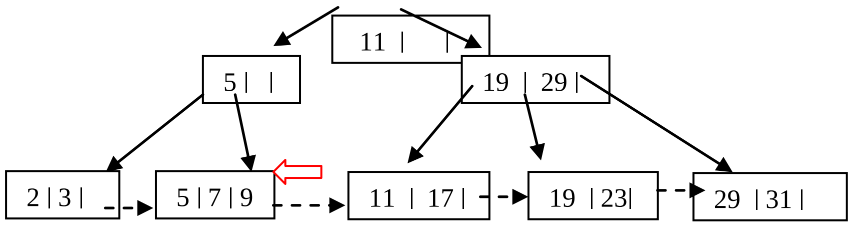
B+tree: Insert 23, 29, 31



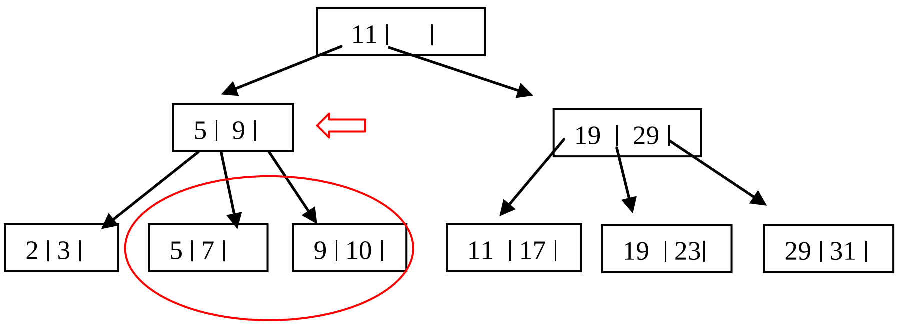
Point 2) B+-tree



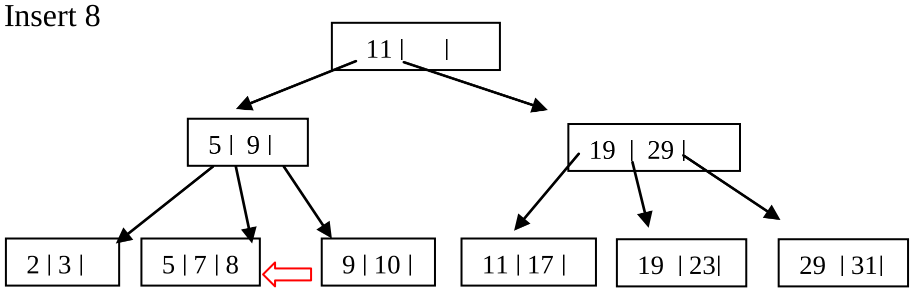
B+tree: Insert 9



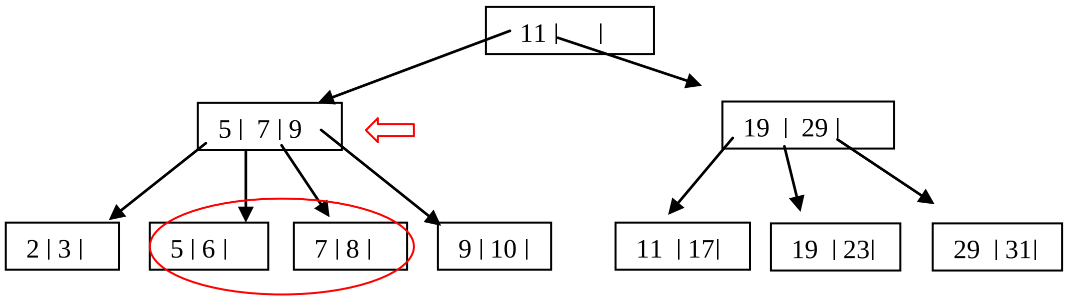
B+tree: Insert 10 (leaf split)



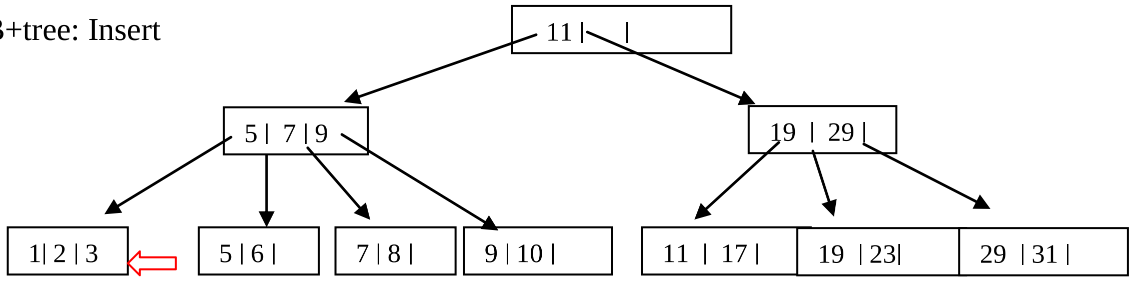
B+tree: Insert 8



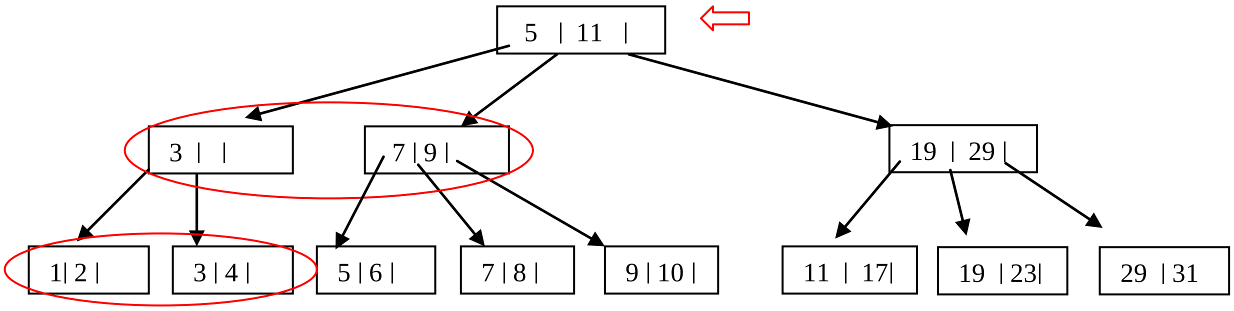
B+tree: Insert 6 (leaf split)



B+tree: Insert



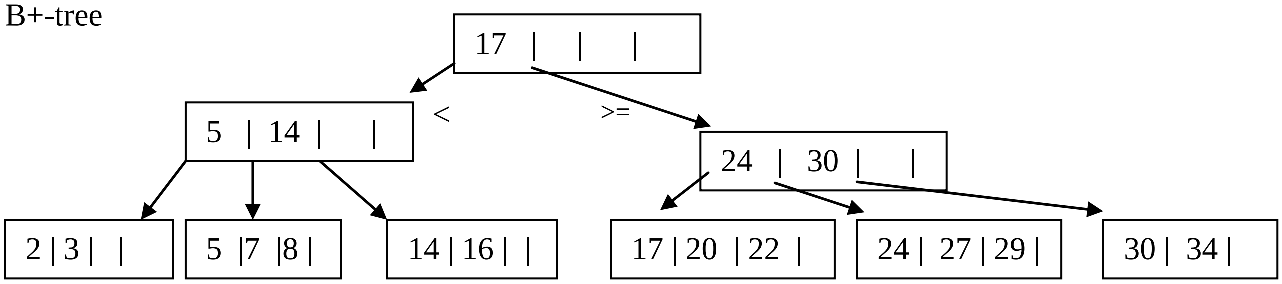
B+tree: Insert 4 (leaf node split + non leaf node split)



EXERCISE 2 **(leaf merge, non leaf merge, leaf keys redistribution)**

For the following B+-tree (m = 5) show the form of the tree after each of the of operations of the sequence:
Delete 17; Delete 20; Delete 34
What is the cost in terms of block transfers for each operation?

B+-tree



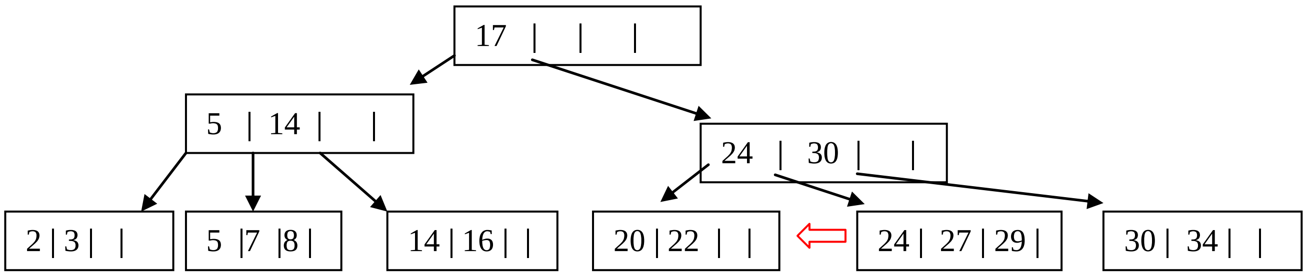
Solution

m=5 Intermediate node : 3<=p<=5 Leaf node: 2<=k<=4
Root: p>= 2 if it is not a leaf ;
0<=k<= 4 if it is a leaf

Delete k:
Find the leaf node that contains k;
Delete k from the node;
If the node has too few entries

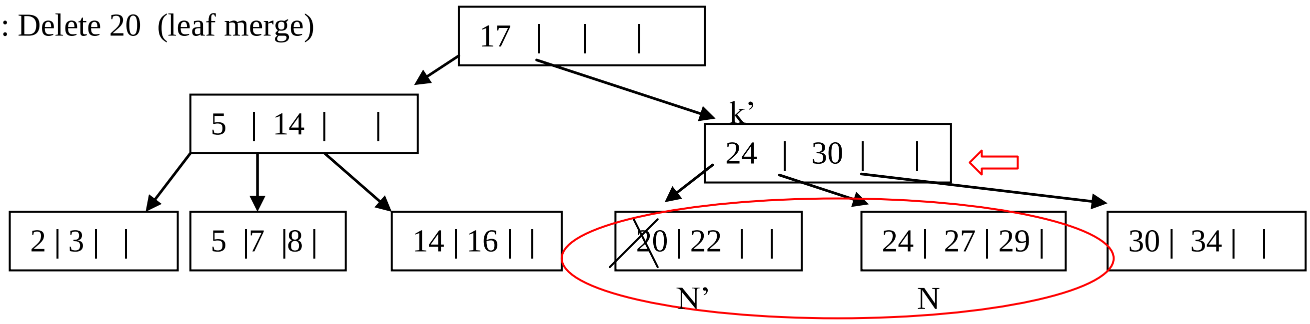
1) merge nodes (if possible)
2) otherwise redistribute keys

B+tree: Delete 17

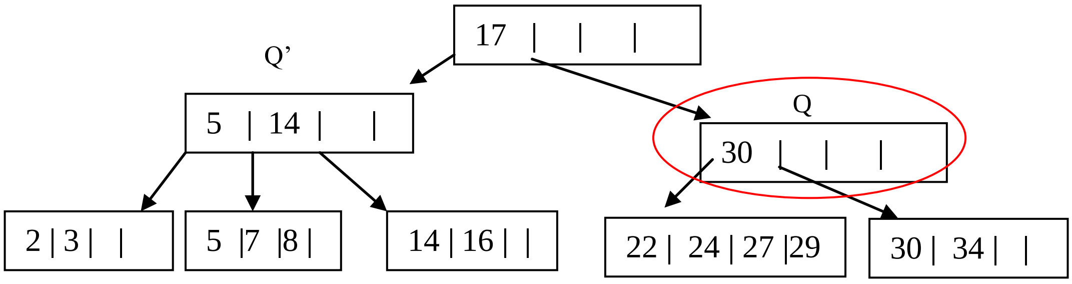


Cost = 3 read + 1 write = 4

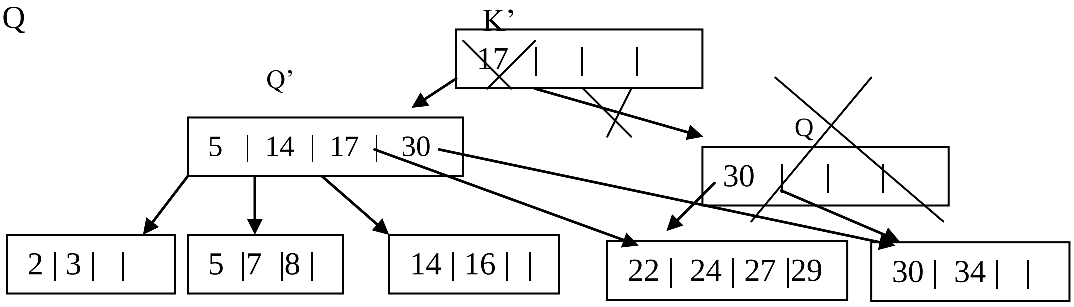
B+tree: Delete 20 (leaf merge)

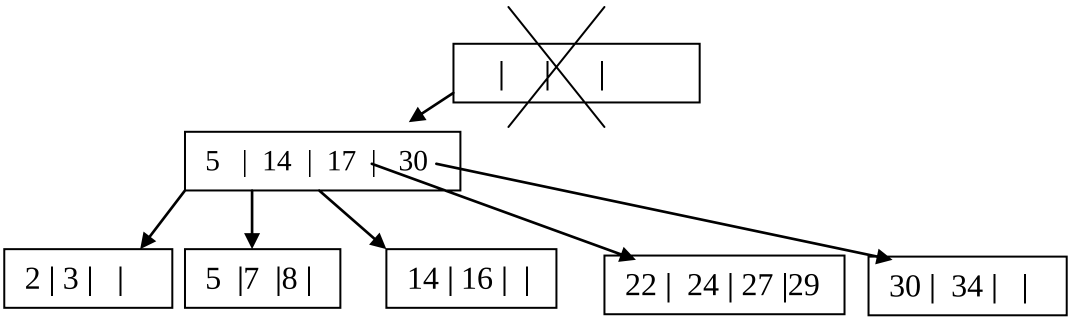


Leaf merge:
Let N' be the predecessor; let N be the successor
Let k' be the value between the two nodes N' and N in the parent
Append all keys to N'
Delete (k, pointer to N) from the parent
Delete N

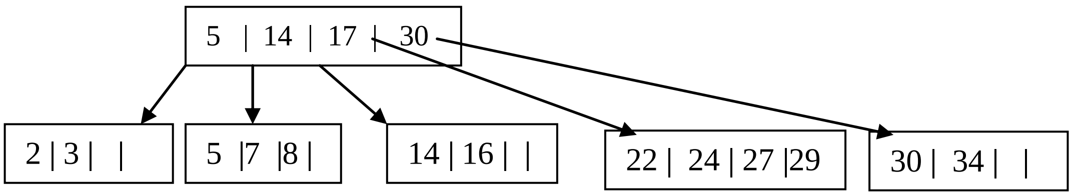


Non leaf merge:
Node Q too few pointers
Let Q' predecessor and Q successor
Let K' be the value between the two nodes in parent of Q
Append K' and all pointers and values of Q to Q'
Delete (K', pointer) from the parent
Delete Q



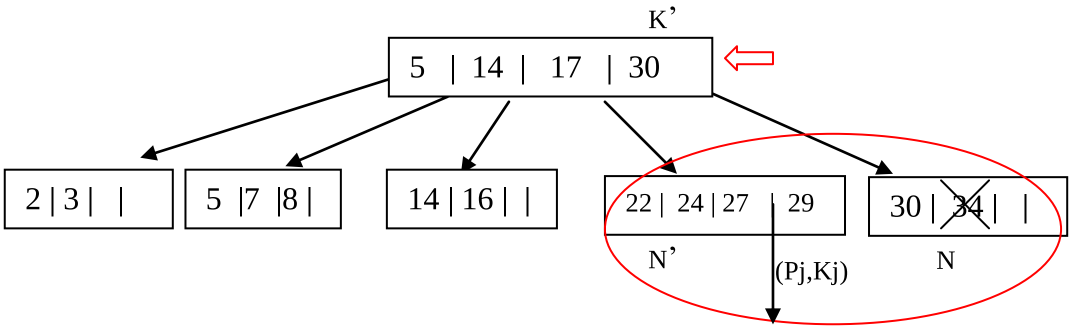


Root has only one child. Root can be deleted.

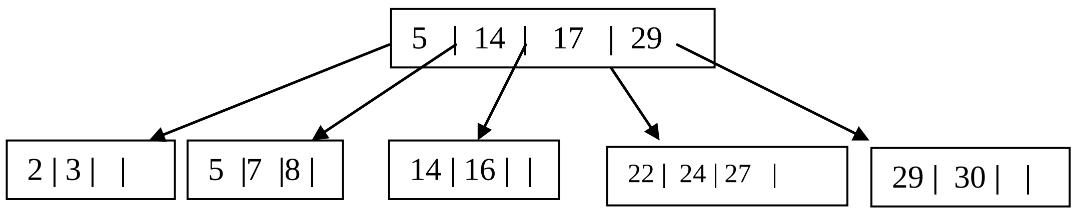


Cost: 5 read + 2 write = 7

Delete 34: keys redistribution (leaf)



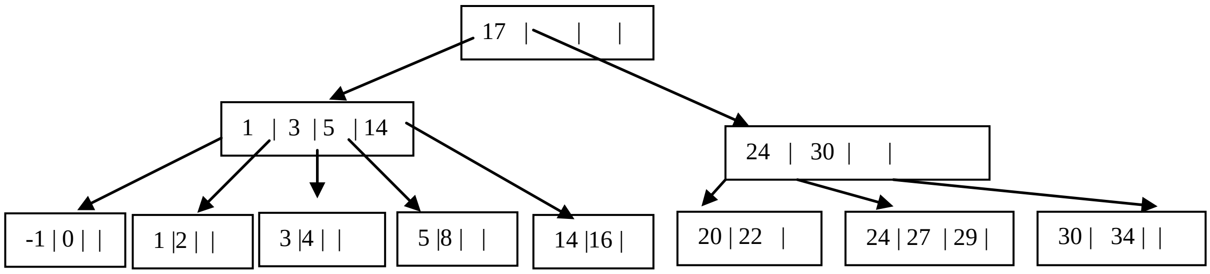
The node N has too few values
Let N' be the previous or next child of parent of N
Let K' the value between N and N' in the parent
Entries of N and N' cannot fit in a single node.
We apply redistribution of keys
N borrows an entry from N' (assume N' predecessor of N)
Let j such that (Pj, Kj) is the last (pointer, value) in N'
Remove (Pj, Kj) from N'
Insert (Pj, Kj) as first value in N
Replace K' by Kj in the parent



Cost = 3 read + 3 write = 6

EXERCISE 3 **(non leaf keys redistribuiton)**

Show the form of the B+-tree (m=5) after the operation Delete 20
What is the cost of the operation?



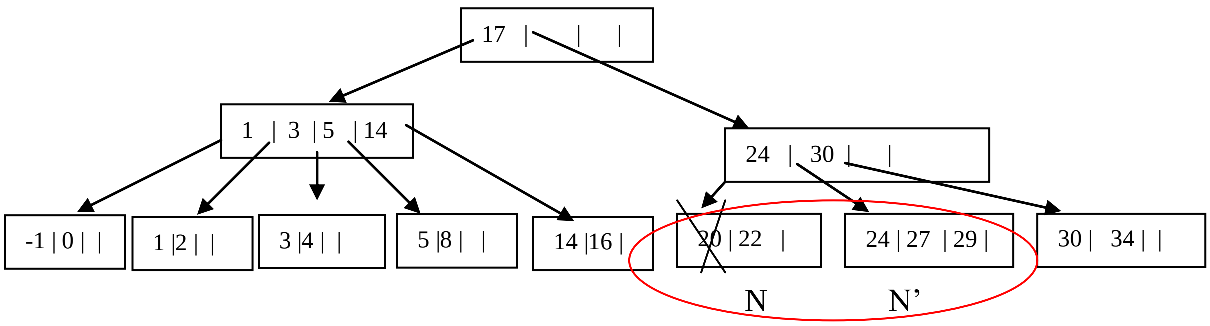
Solution

m=5 Intermediate node : 3<=p<=5 Leaf node: 2<=k<=4

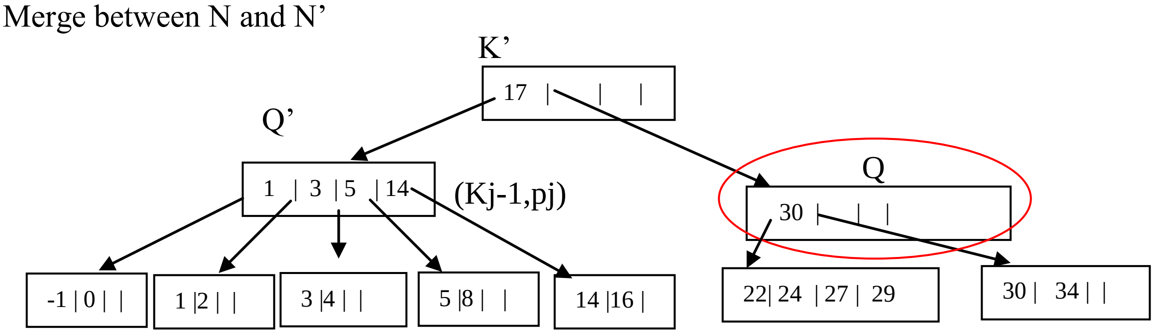
Root: p>= 2 if it is not a leaf ;

0<=k<= 4 if it is a leaf

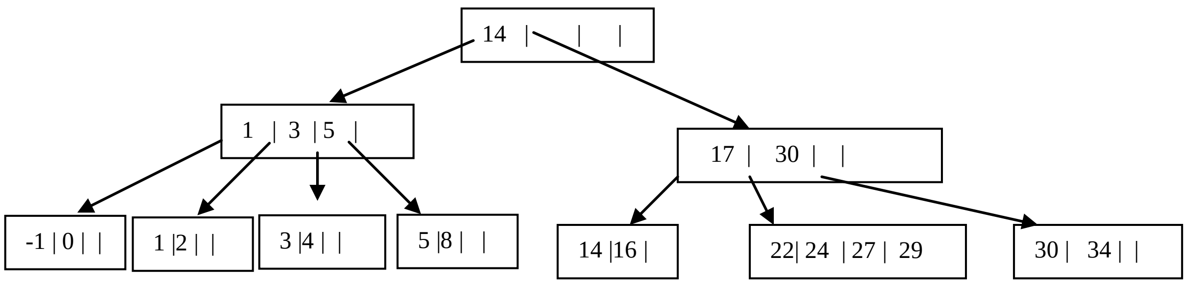
Delete 20



The node N has too few values
Merge between N and N'



The node Q has too few pointers
No merge with previous or net child of parent of Q
Redistribution of keys
Q borrows an entry from Q'
Let j be such that (Kj-1, pj) is the last value pointer in Q'
Let K' the value between Q and Q' in parent of Q
Insert (pj, K') as first value of Q
Remove (Kj-1, pj) from Q'
Replace K' with Kj-1 in parent of Q



Cost= 5 read + 4 write = 9