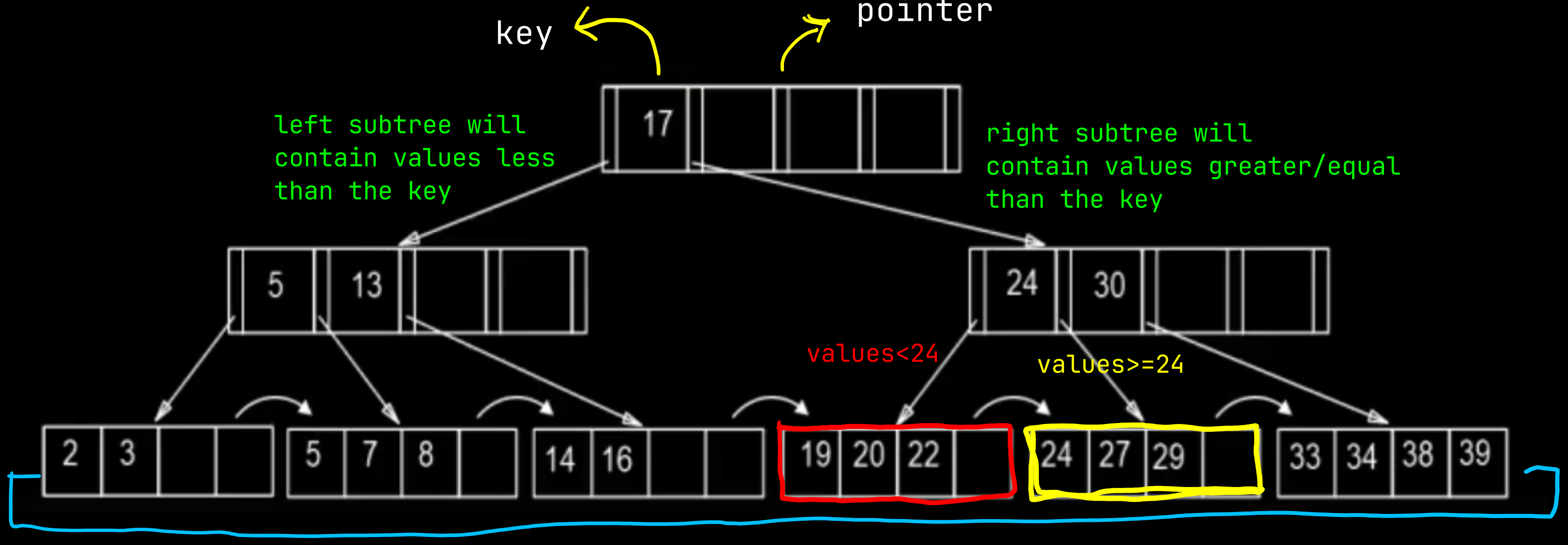
B+ Tree



key

pointer

left subtree will contain values less than the key

right subtree will contain values greater/equal than the key

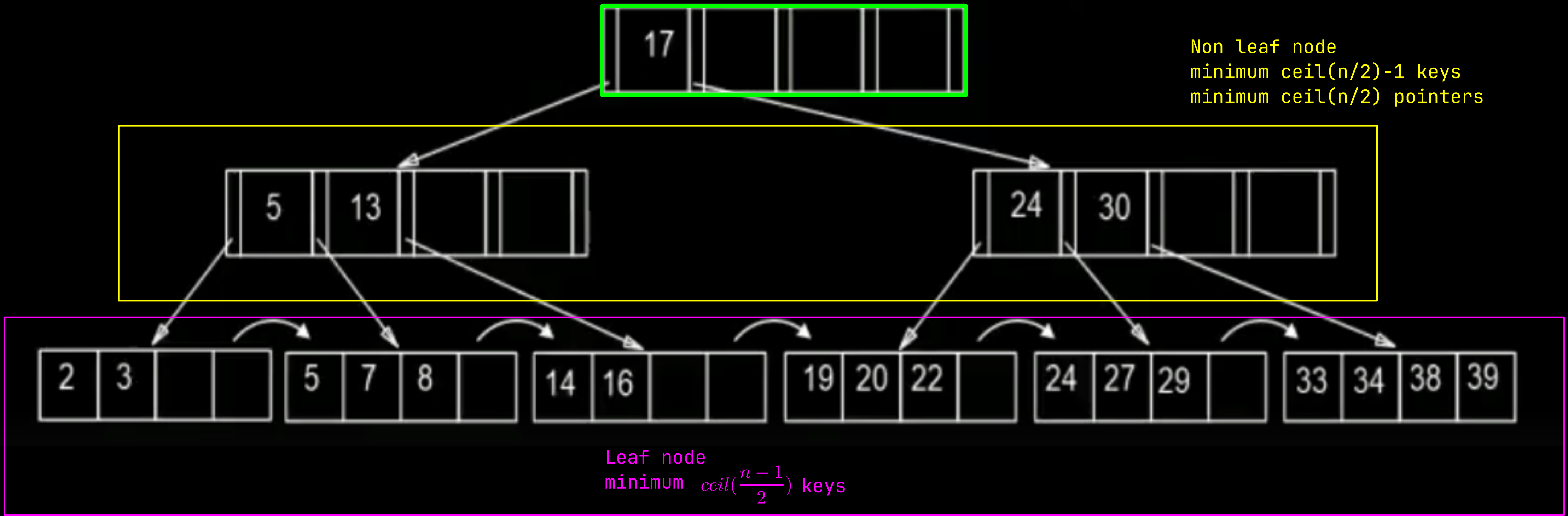
values<24

values>=24

sorted

Root node
minimum 1 key
minimum 2 pointer

Non leaf node
minimum ceil(n/2)-1 keys
minimum ceil(n/2) pointers



Leaf node
minimum $ceil(\frac{n-1}{2})$ keys

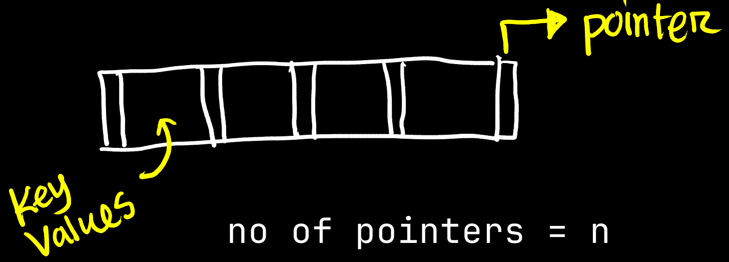
minimum $ceil(\frac{n-1}{2})+1$ pointers

Root-> leaf (no child/pointer) -> minimum 0 keys
-> non leaf -> minimum 2 children

Maximum is same for all node
maximum n-1 key
maximum n pointers

Properties

1. Degree/Order/no of maximum child/no of pointers, n = 5



no of pointers = n
no of key = n-1

2. Ordered Data Structure : always sorted in a level

3. Dynamic Tree : automatically adjust height

Consider a B+ tree of order 5. now push these

10, 50, 26, 13, 17, 24, 31, 3, 29, 42, 9, 62

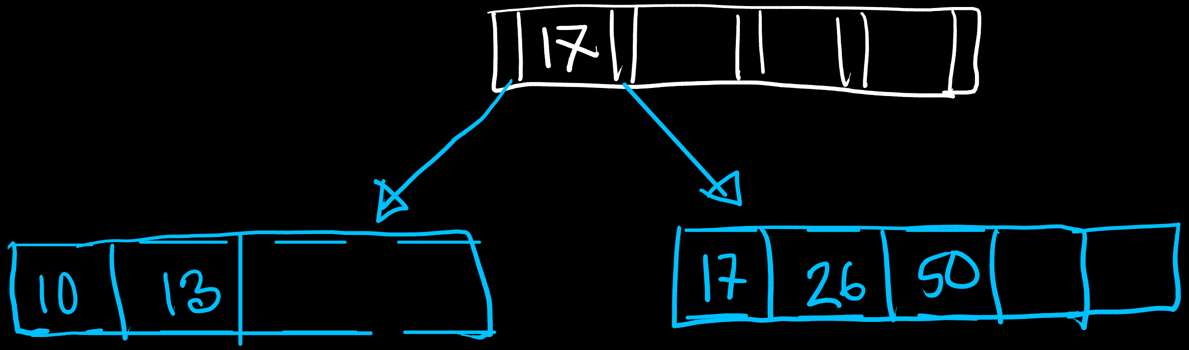
order 5 -> no of pointers = 5
no of keys = 4

inserting 10,50,26,13

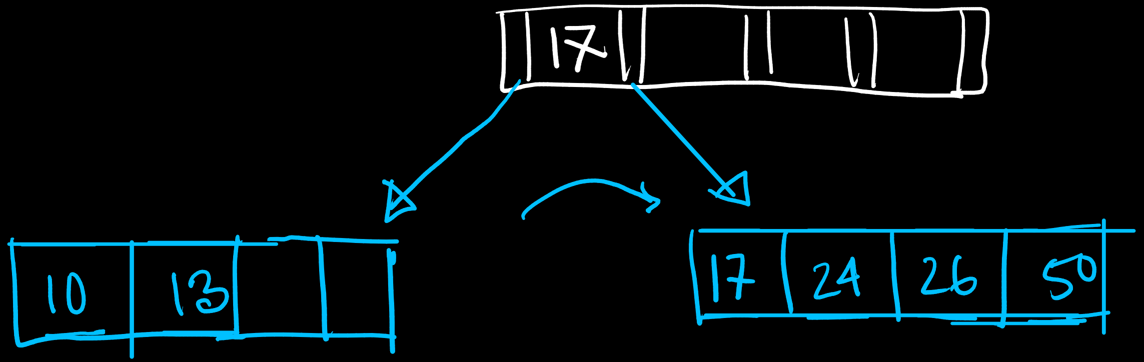
10	13	26	50

10 13 17 26 50

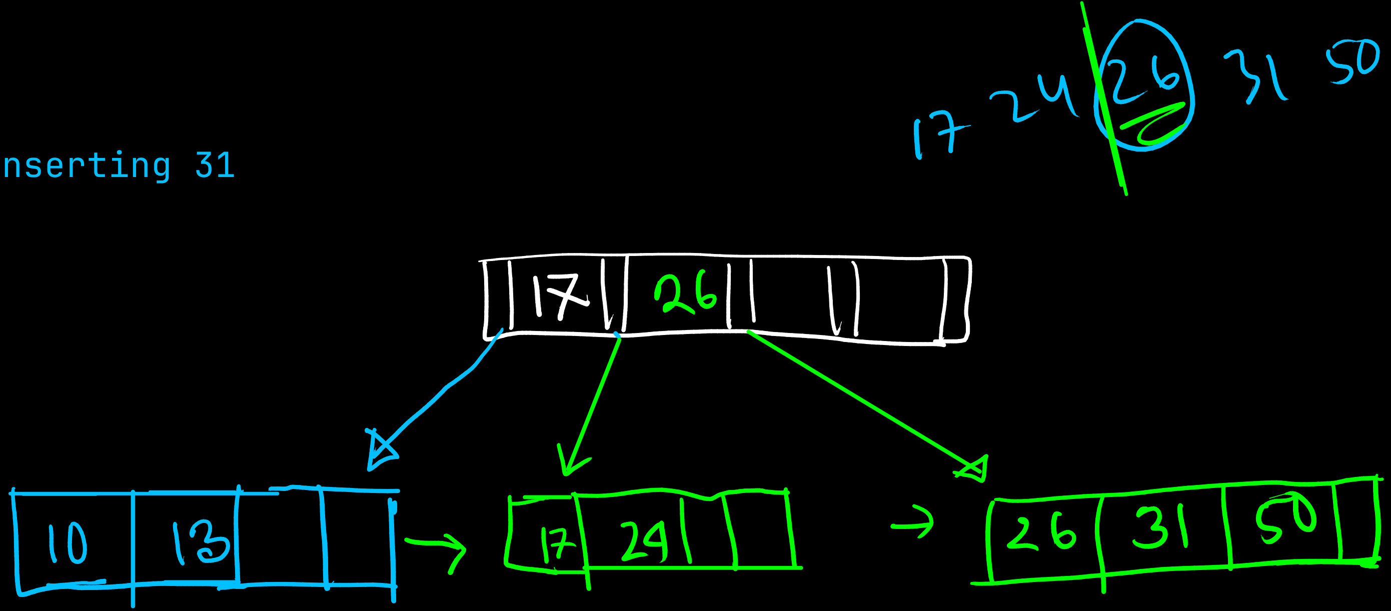
inserting 17



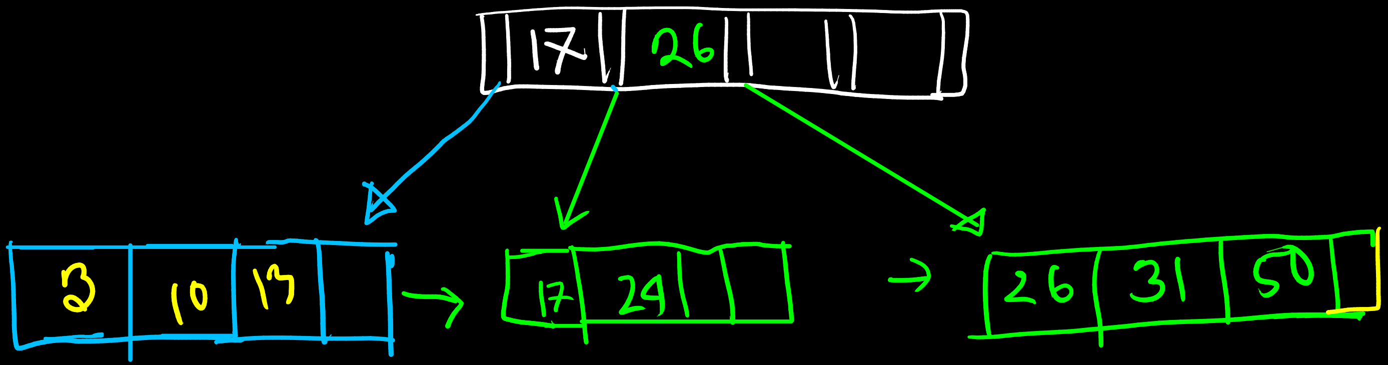
inserting 24



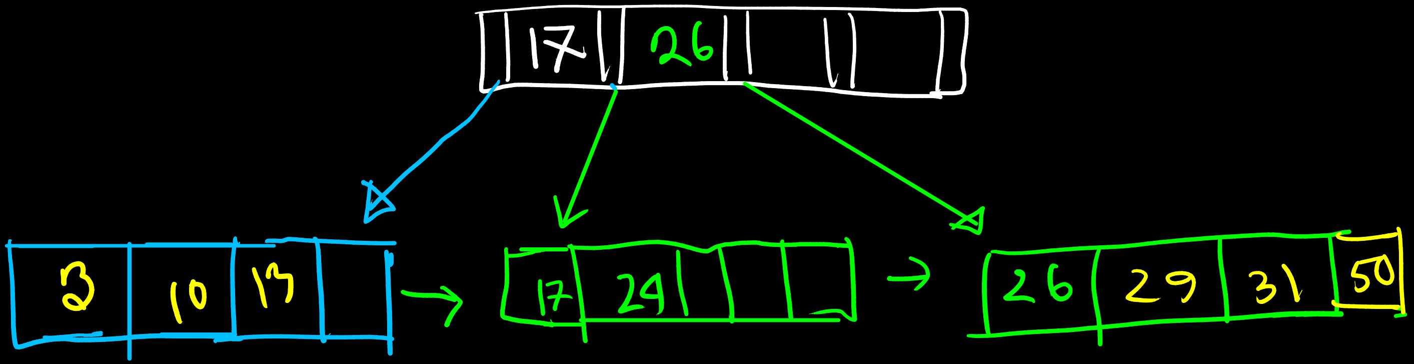
inserting 31



inserting 3

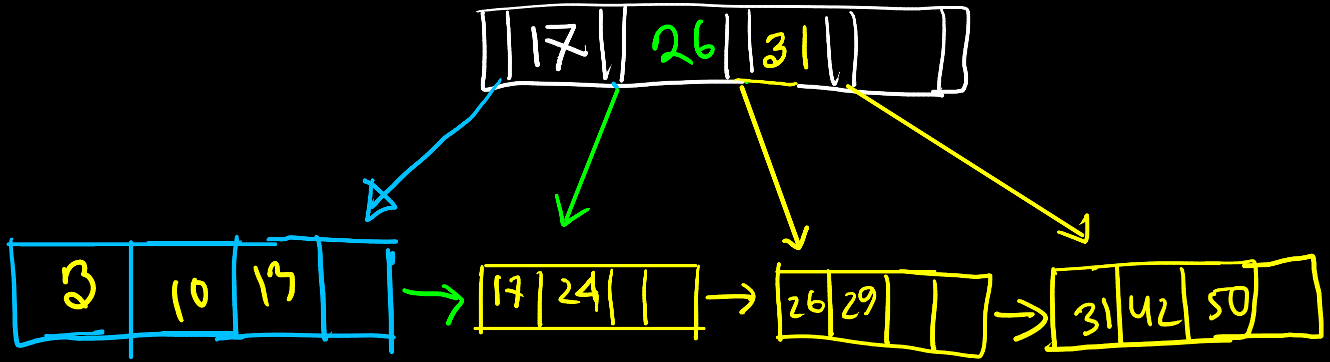


inserting 29

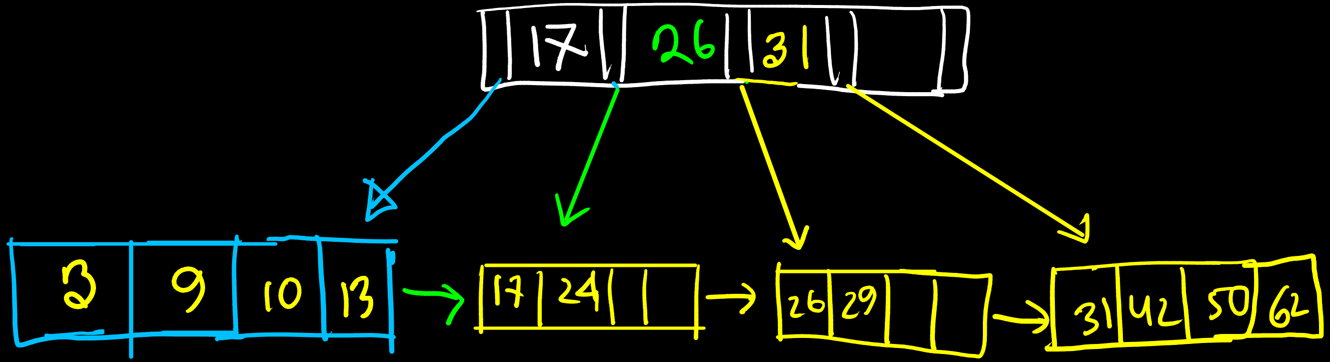


inserting 42

26 29 31 42 50



inserting 9, 62


17
26
31
3
9
10
13
17
24
26
29
31
42
50
62