

02/23/09

B+ tree Insertion & Deletion.

Invariants

1. Each node has $F \in [\lceil L/2 \rceil, d]$
2. Each node has F key values, $F+1$ pointers pointing to $F+1$ children.

Insertion

Find the leaf node L to insert
currentnode $\leftarrow L$; ($\rightarrow d$ key values)
while (currentnode is NOT valid) {
 split (currentnode);
 currentnode \leftarrow currentnode.parent;
}

split (N) {

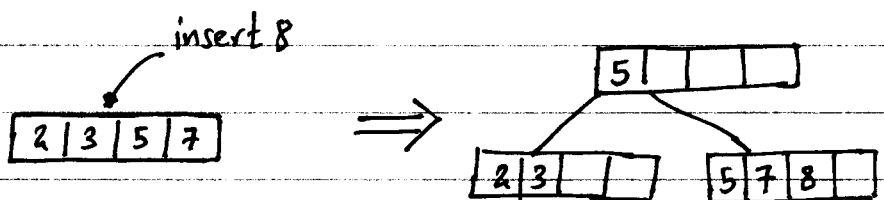
 create new node M ;

 evenly distribute all $d+1$ keys into N and M ;

 promote the middle key to parent;

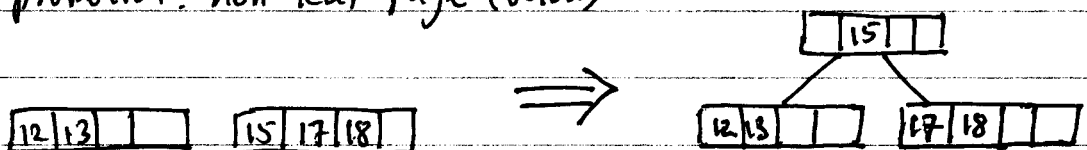
}

Example: $d=4$

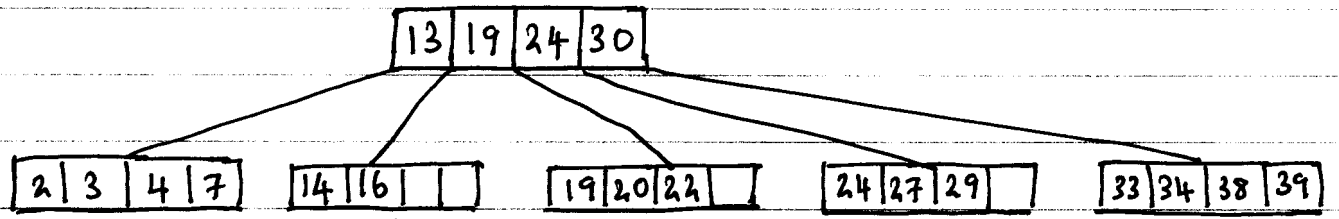


case 1 of promotion: leaf page do copy up (above)

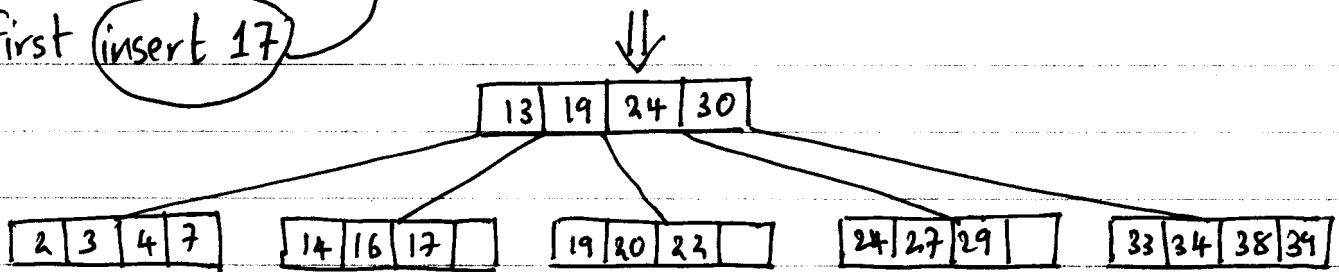
case 2 of promotion: non-leaf page (below)



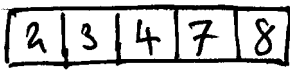
Example: $d=4, F=[2,4]$



first (insert 17)



Then (insert 8)



↓ split

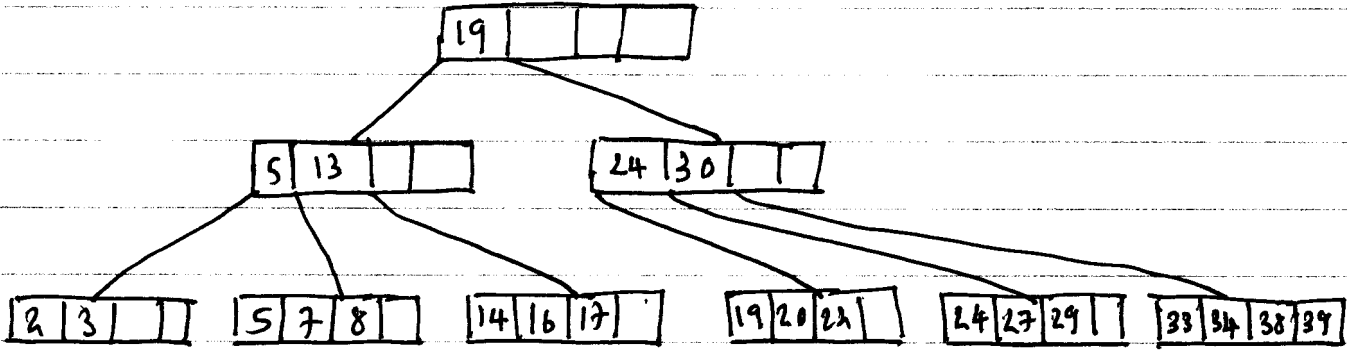


then [5 | 13 | 19 | 24 | 30]

↓ split



⇓



Deletion:

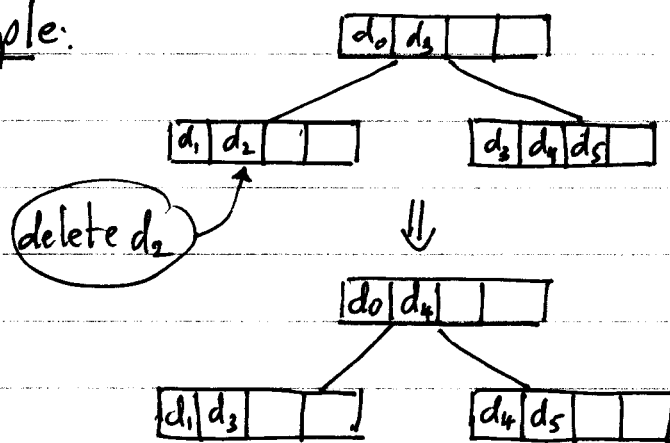
```
currentnode ← L;  
while (currentnode is NOT valid) {  
  redistribute;  
  if redistribute fails  
    merge (currentnode);  
    currentnode ← currentnode.parent;  
  else return;  
}
```

($< \lfloor \frac{d}{2} \rfloor$ key values)

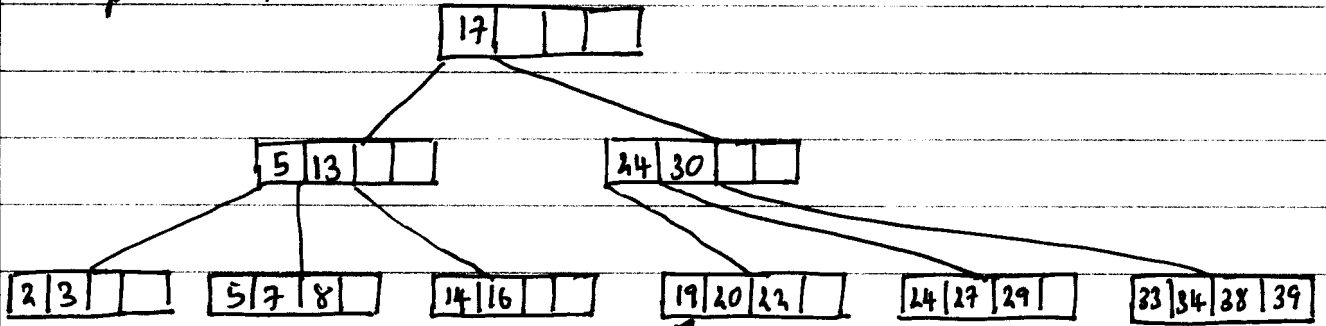
redistribute

borrow data from a sibling

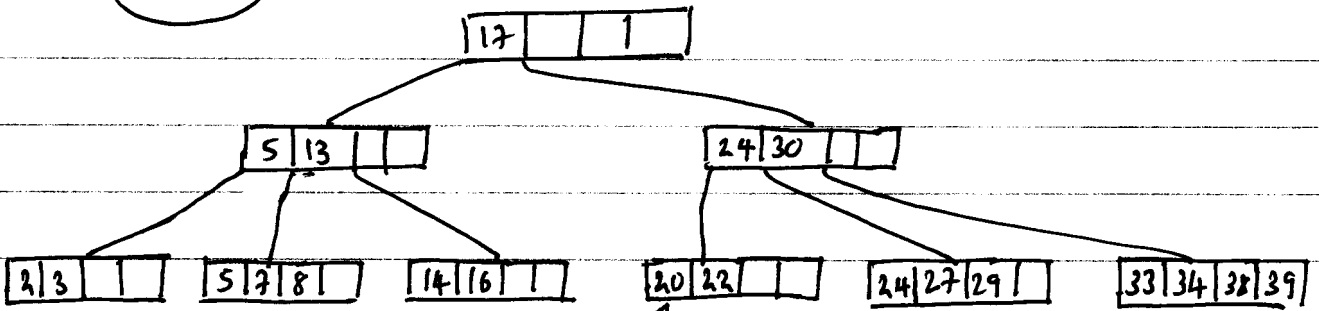
Example:



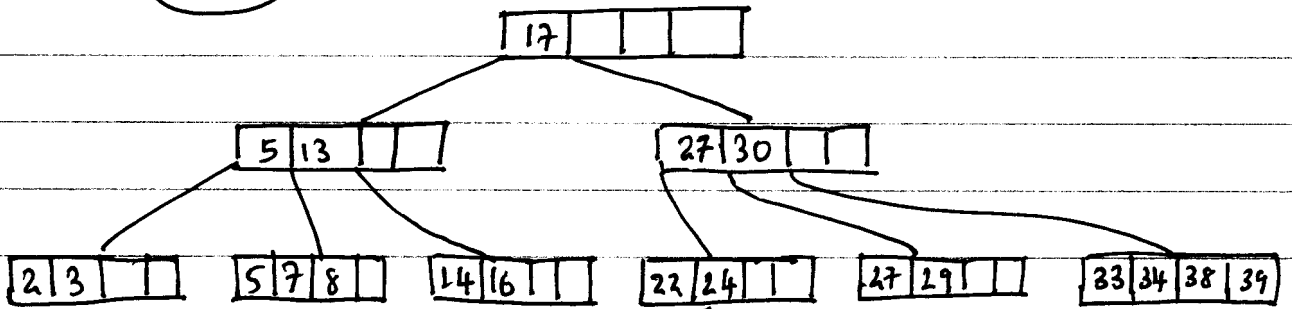
Example: $d=4$



First delete 19



Second delete 20



Third delete 24

