

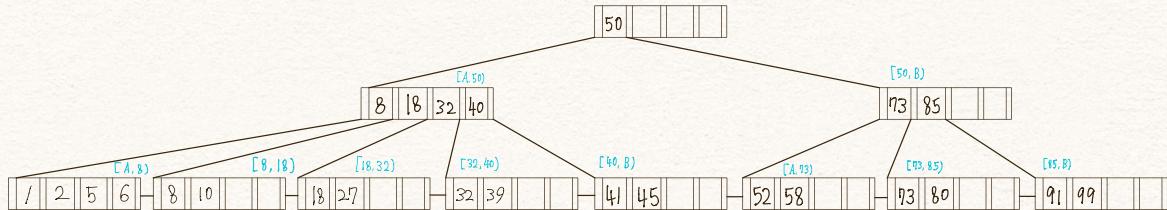
B + Tree Indexing

B+ tree w/ order: 2

min # of keys
in a node

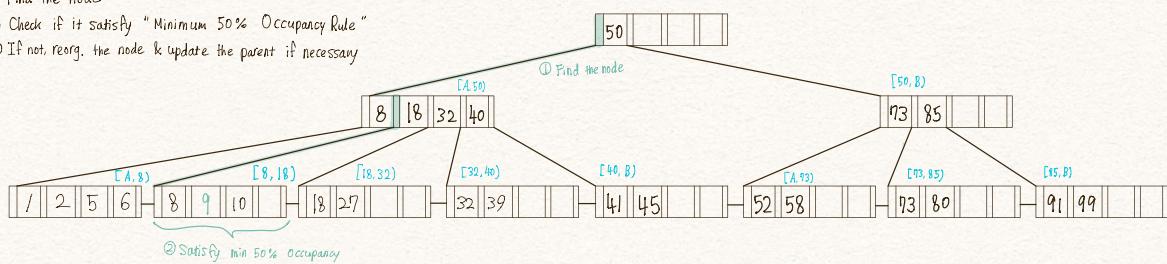
Original B+ Tree:

Each node can have $d \sim 2d$ key values ranging from $[0, \square)$
 $2d+1$ pointers

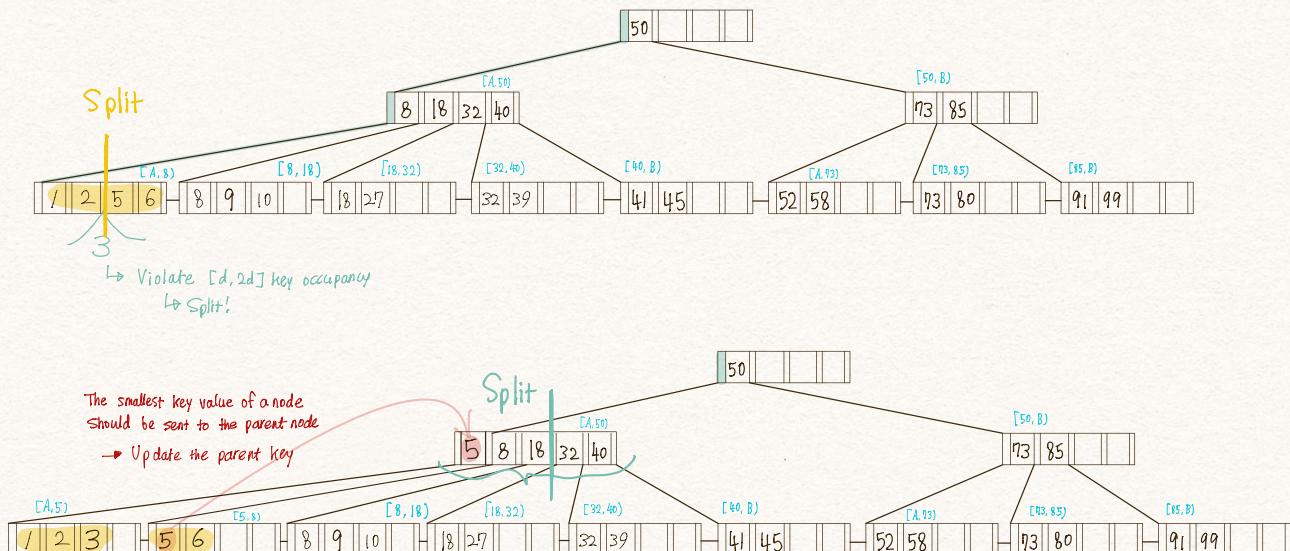


Q1. Insertion: a key "9"

- ① Find the node
- ② Check if it satisfy "Minimum 50% Occupancy Rule"
- ③ If not, rearr. the node & update the parent if necessary

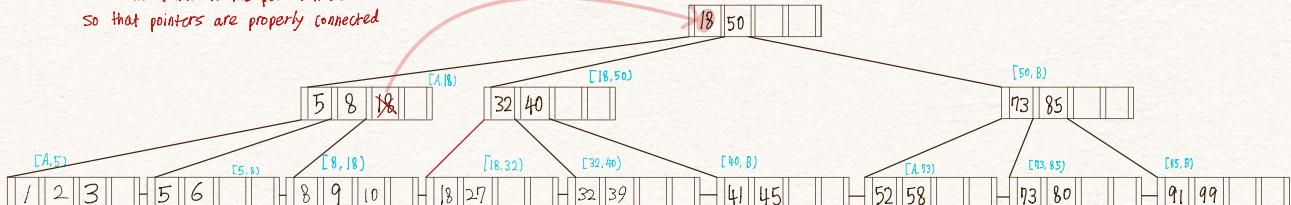


Q2. Insertion: a key "3"



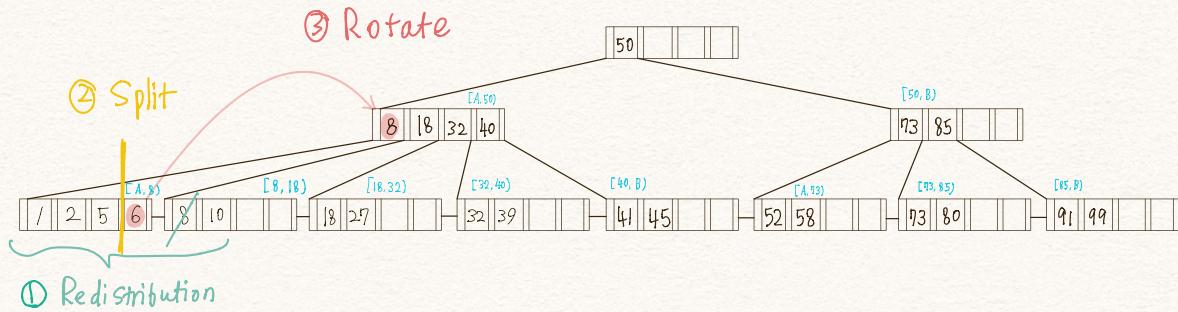
Send the value to the parent node
so that pointers are properly connected

Rotate

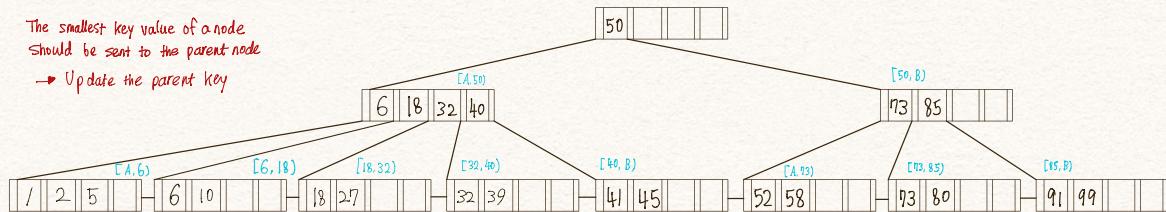


From the original B+ tree:

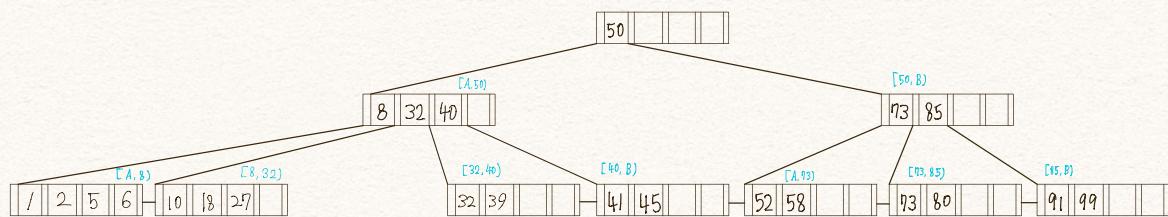
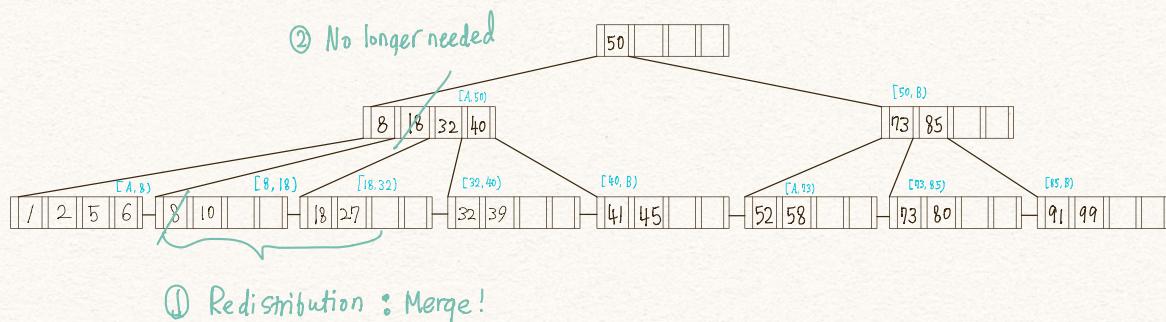
Q3. Deletion: a key "8" → Redistribution w/ the "left" sibling



The smallest key value of a node
Should be sent to the parent node
→ Update the parent key

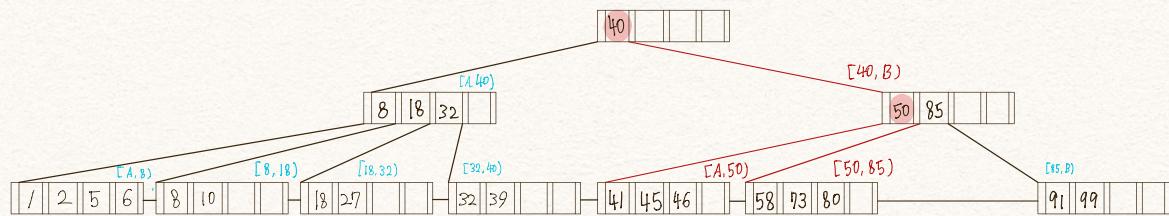
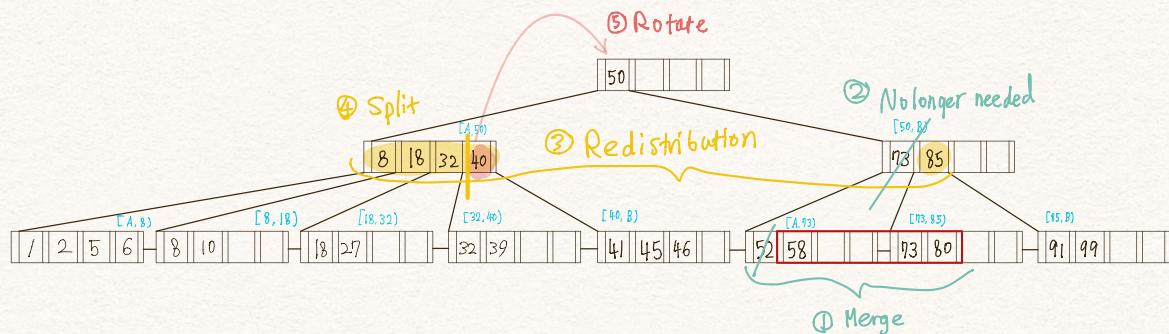


Q4. Deletion: a key "8" → Redistribution w/ the "right" sibling



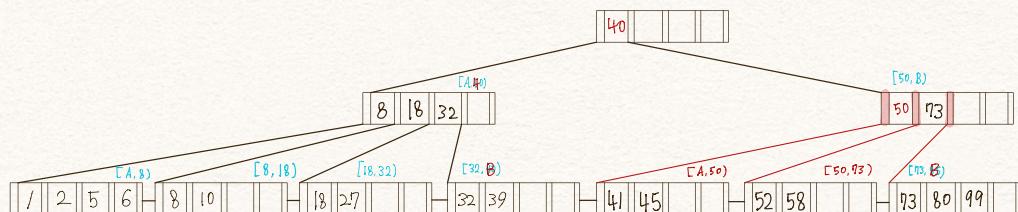
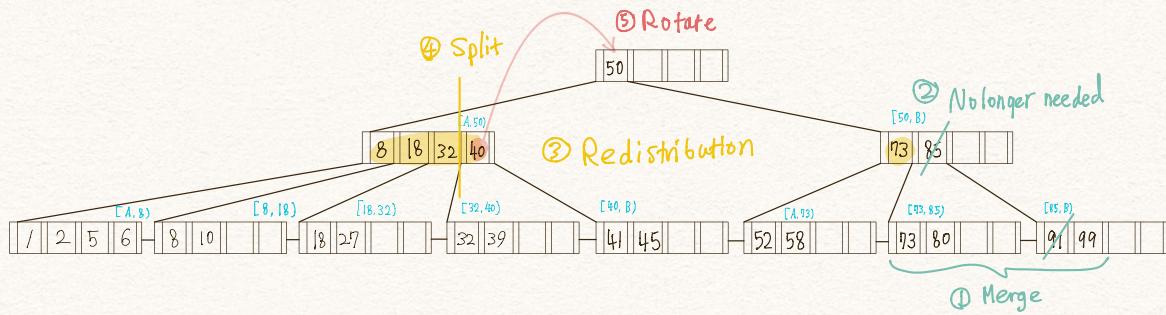
From the original B+ tree:

Q5. Insertion of key "46" & Deletion of key "52"



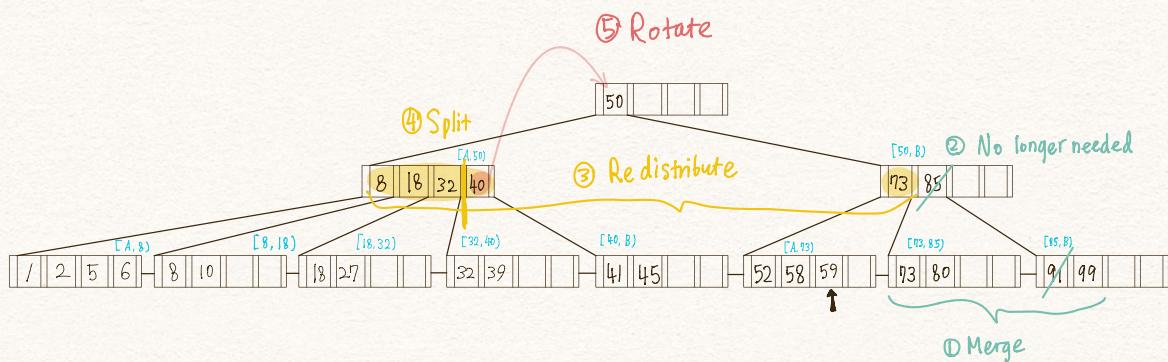
From the original B+ tree:

Q6. Deletion of a key "91"



From the original Bt tree:

Q7. Insertion of key "59" and Deletion of key "91"



From the original Bt tree:

Q8. Deletion of keys 32, 39, 41, 45, and 73

