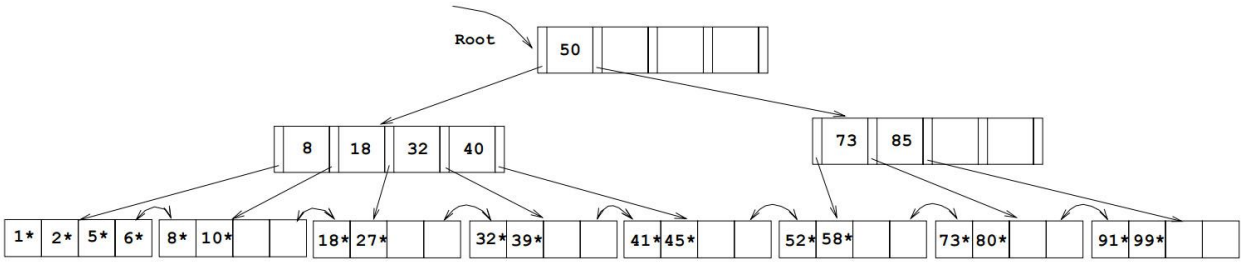
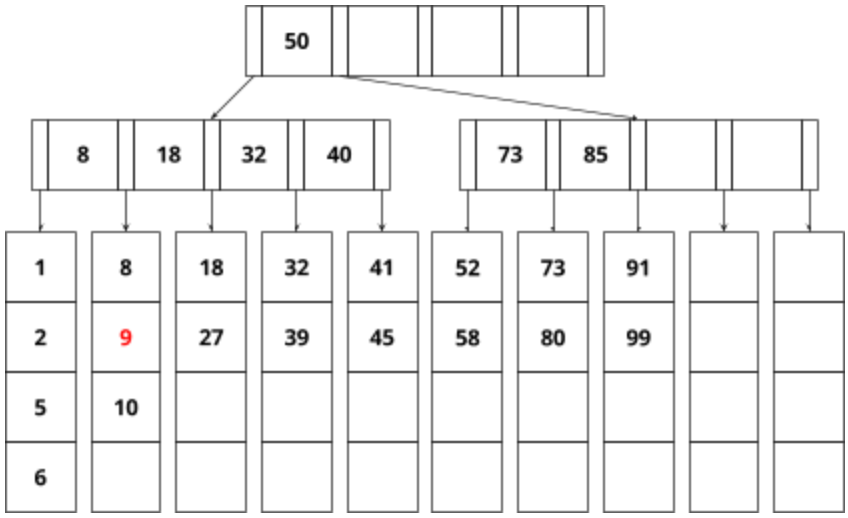
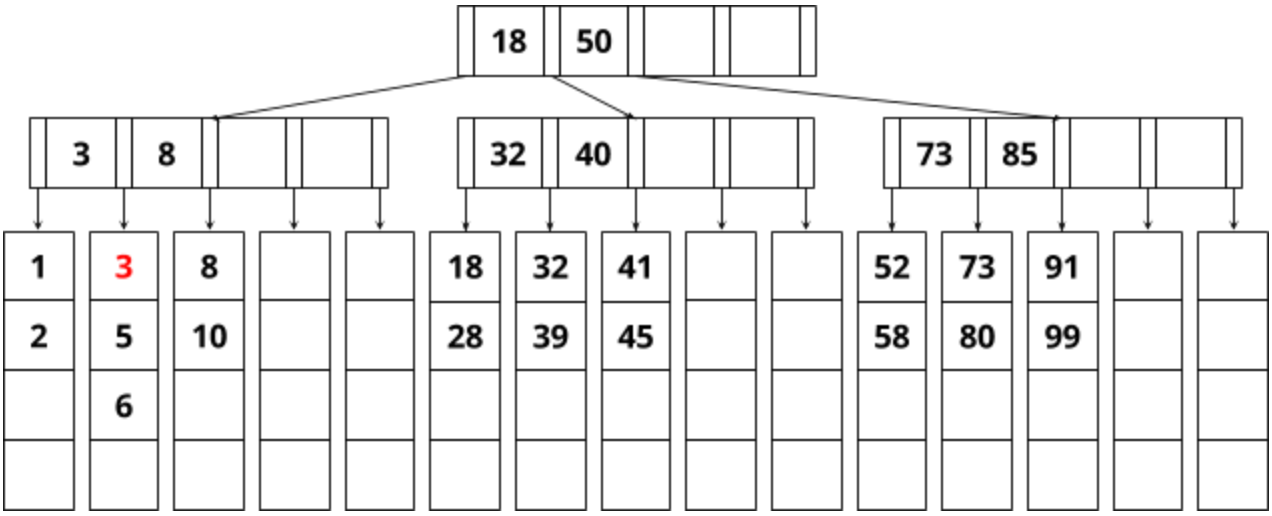
1. **Exercise 10.1** Consider the B+ tree index of order d = 2 shown below.



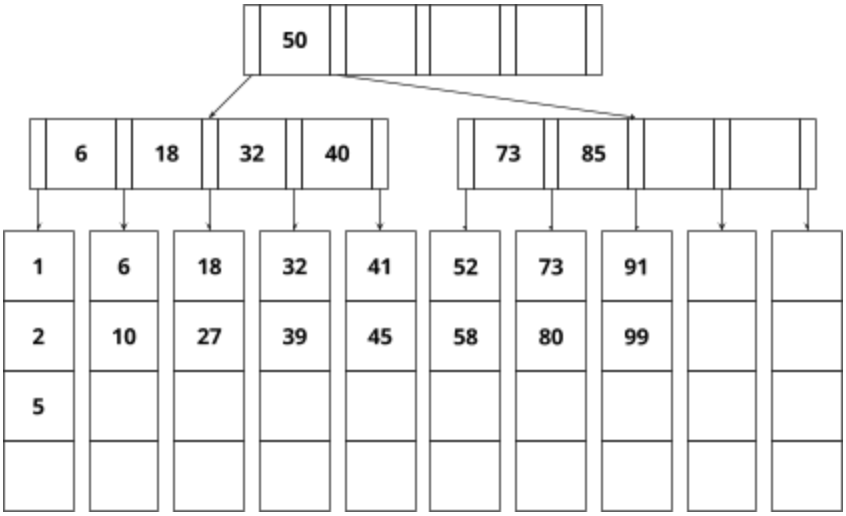
   1. Show the tree that would result from inserting a data entry with key 9 into this tree.



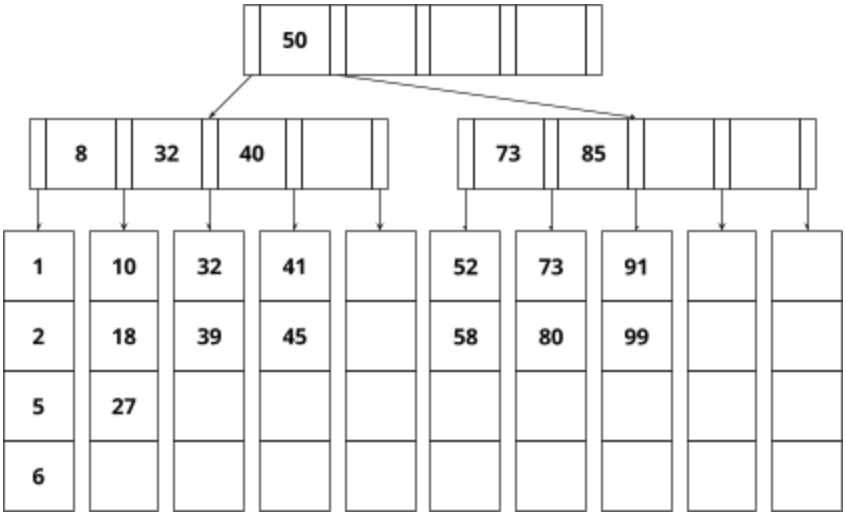
   2. Show the B+ tree that would result from inserting a data entry with key 3 into the original tree. How many page reads and page writes does the insertion require?



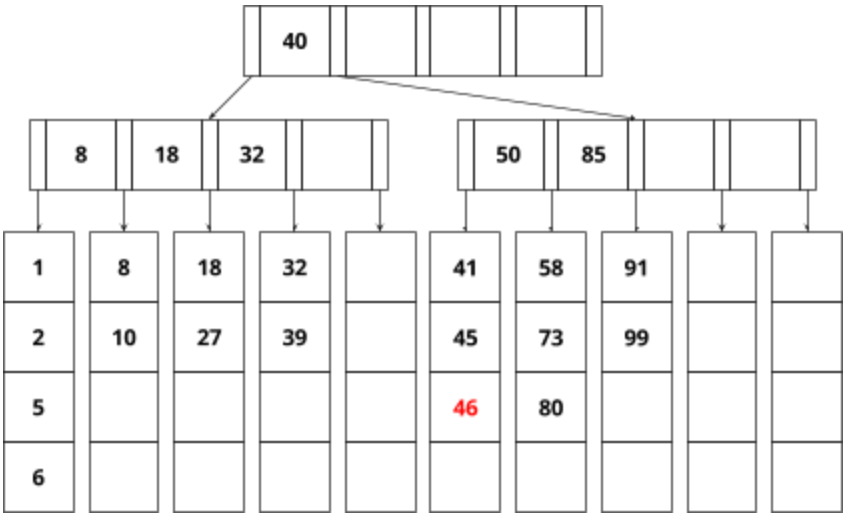
3. Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the left sibling is checked for possible redistribution.



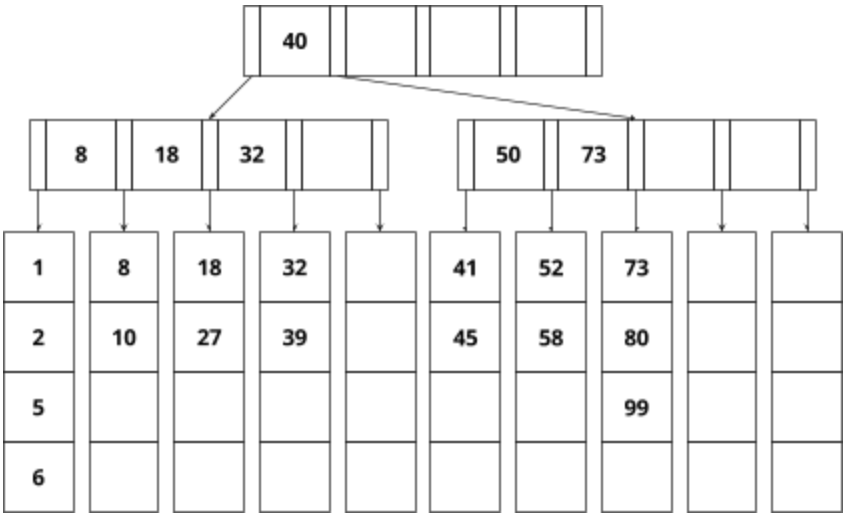
4. Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the right sibling is checked for possible redistribution.



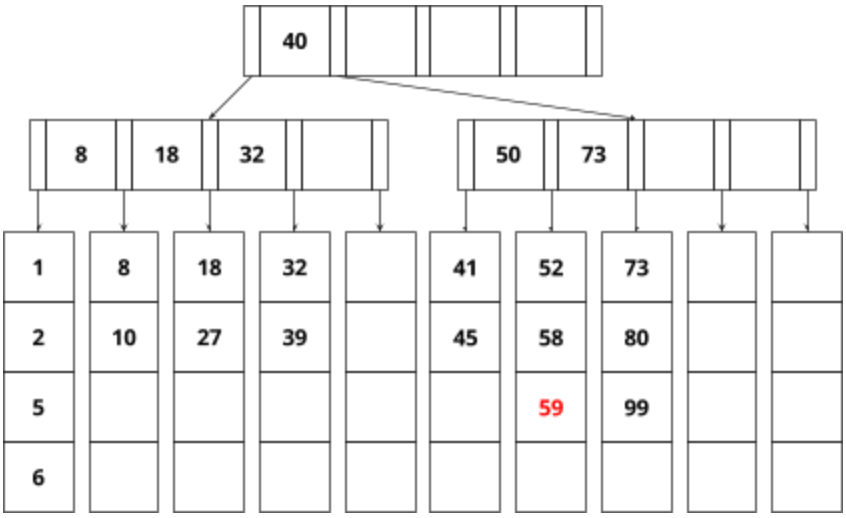
5. Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 46 and then deleting the data entry with key 52.



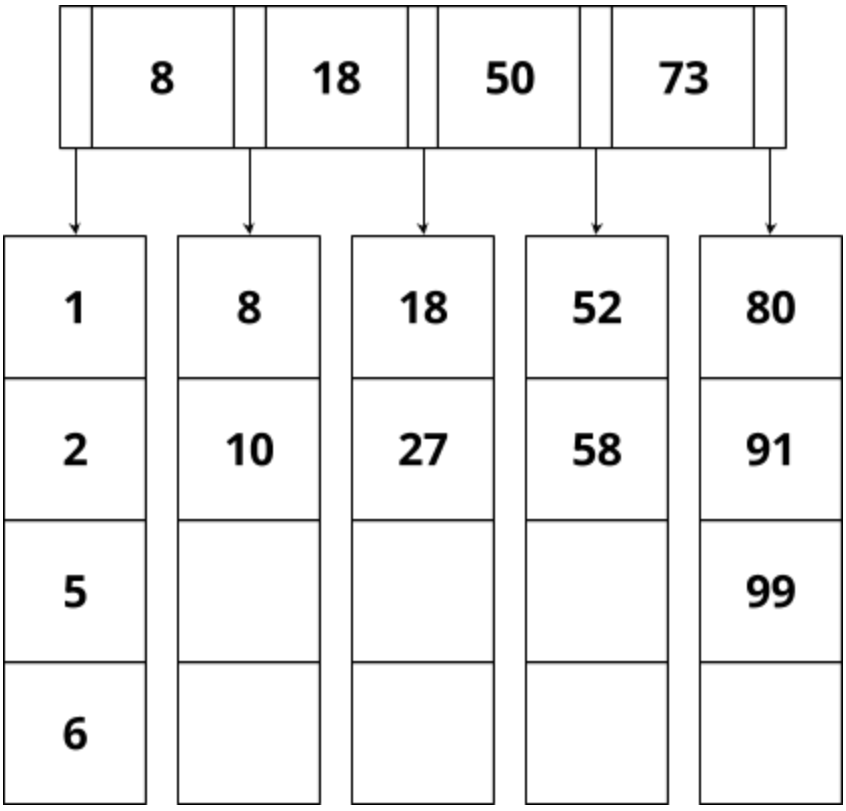
6. Show the B+ tree that would result from deleting the data entry with key 91 from the original tree.



7. Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 59, and then deleting the data entry with key 91.



8. Show the B+ tree that would result from successively deleting the data entries with keys 32, 39, 41, 45, and 73 from the original tree.



2. Construct a B+-tree for the following set of key values: 29, 19, 47, 43, 70, 21, 57, 36, 14, 78, 53, 13, 101, 64, 72, 105. Assume the tree is initially empty and the values are added in the order specified by the above list. The maximum number of search key values that will fit in one node is four. When splitting an overflowing node with an odd number of values, you always put one more value in the right hand side node.