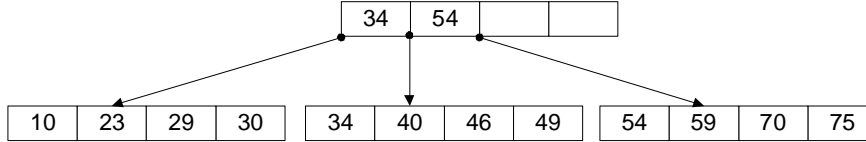


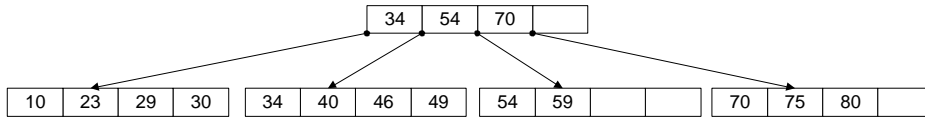
Section solutions
12 Feb 2003

B+Tree

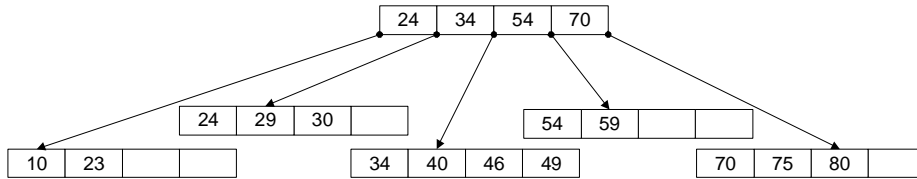
- 1) Bulk load the B+Tree with values 46, 10, 70, 23, 40, 59, 29, 34, 54



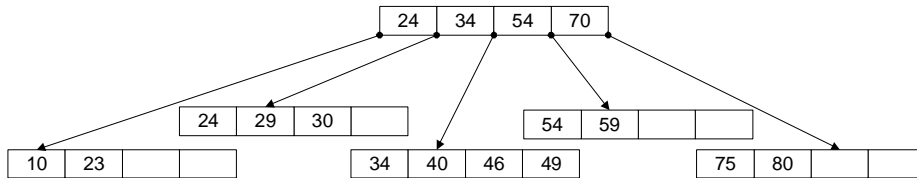
- 2) Insert 80



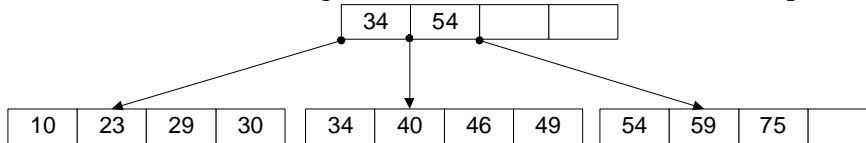
- 3) Insert 24



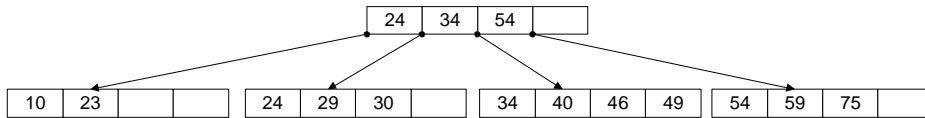
- 4) Remove 70



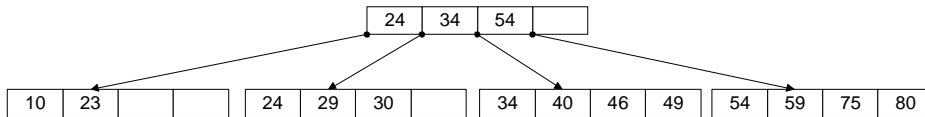
- 5) Go back to the original bulk loaded B+Tree (step 1). Remove 70



- 6) Insert 24



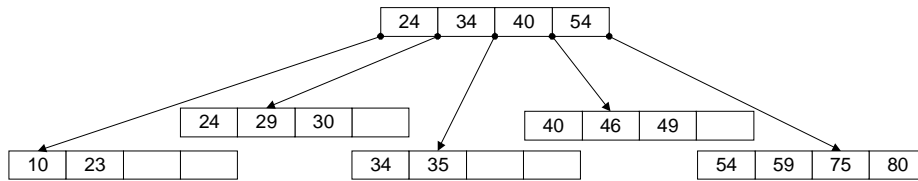
- 7) Insert 80



Notice: although the index has the same leaves as step 4, its organization is different because of the order of operations.

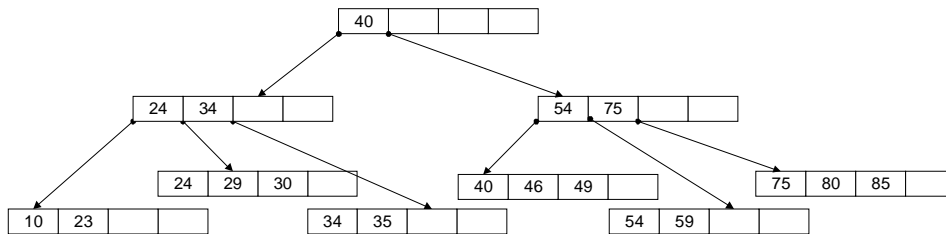
8) Insert

35



9) Insert

85



10) Leaf node with {54} runs below 2 entries. Redistribute with {75, 80, 85} by moving 75 over to the left, and replacing 75 with 80 for the parent index entry. The two leaf nodes are now {54, 75} and {80, 85}.

11) We can do redistribution with the left node. Suppose the left node has less than 2 entries, we will have to merge. In this case, we merge {54, 75} and {80, 85}. This leads to removal of the index entry {80}, which causes the two index entries to be merged, and the root node {40} to be *pulled down*. The root is now {24, 34, 40, 54}

12-15) All answers are 2! This is one of the benefits of B+Trees, very predicable!

External Sorting

- 1) Each page can store up to 10 records of 48 bytes each. 450 pages are needed to store 4500 records.
- 2) Number of sorted runs = $\text{ceil}(450/4) = 113$ of 4 pages each, except the last run which is only 2 pages long.
- 3) The total number of passes = $\log_3(113) + 1 = 6$ passes.
- 4) The total IO cost for sorting the file = $2 * 450 * 6 = 5400$ I/Os.
- 5) Let N be the number of pages. Let B be the number of buffers. $N = B * (B-1)$. When $B=3$, $N = 12$. When $B=257$, $N=65792$. The number of records = $N * 10$.
- 6) Tournament sort will result in runs that are $2B$ long. E.g., if $B=4$, then we can have up to 8 pages per run, which means that the number of sorted runs in step 2 is $\text{ceil}(450/8)$. You should be able to work out the rest by yourself.