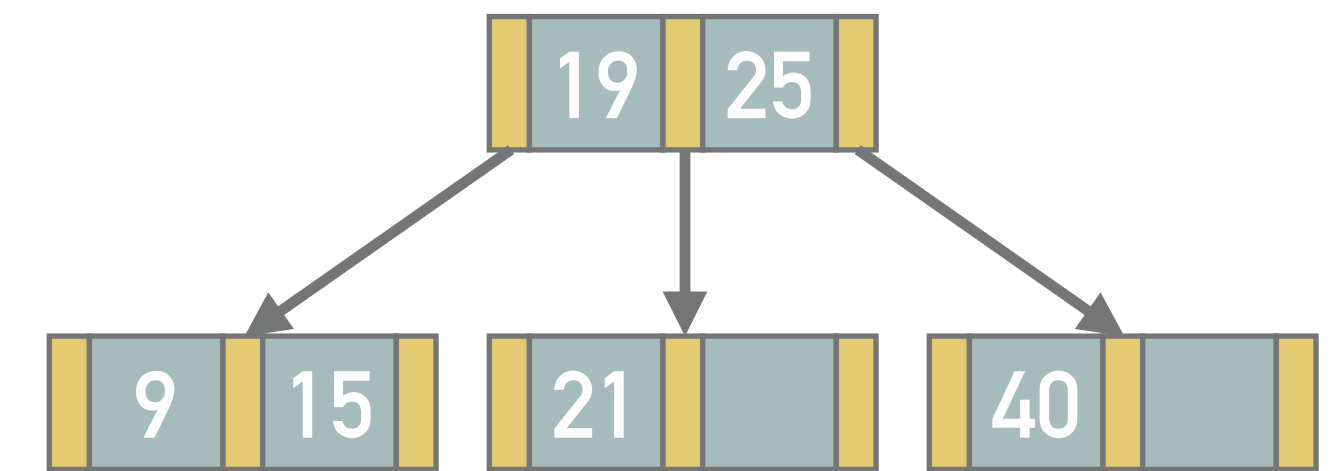


B-TREES: SOLUTION

NDBI007: Practical Class 5

EXERCISE 1

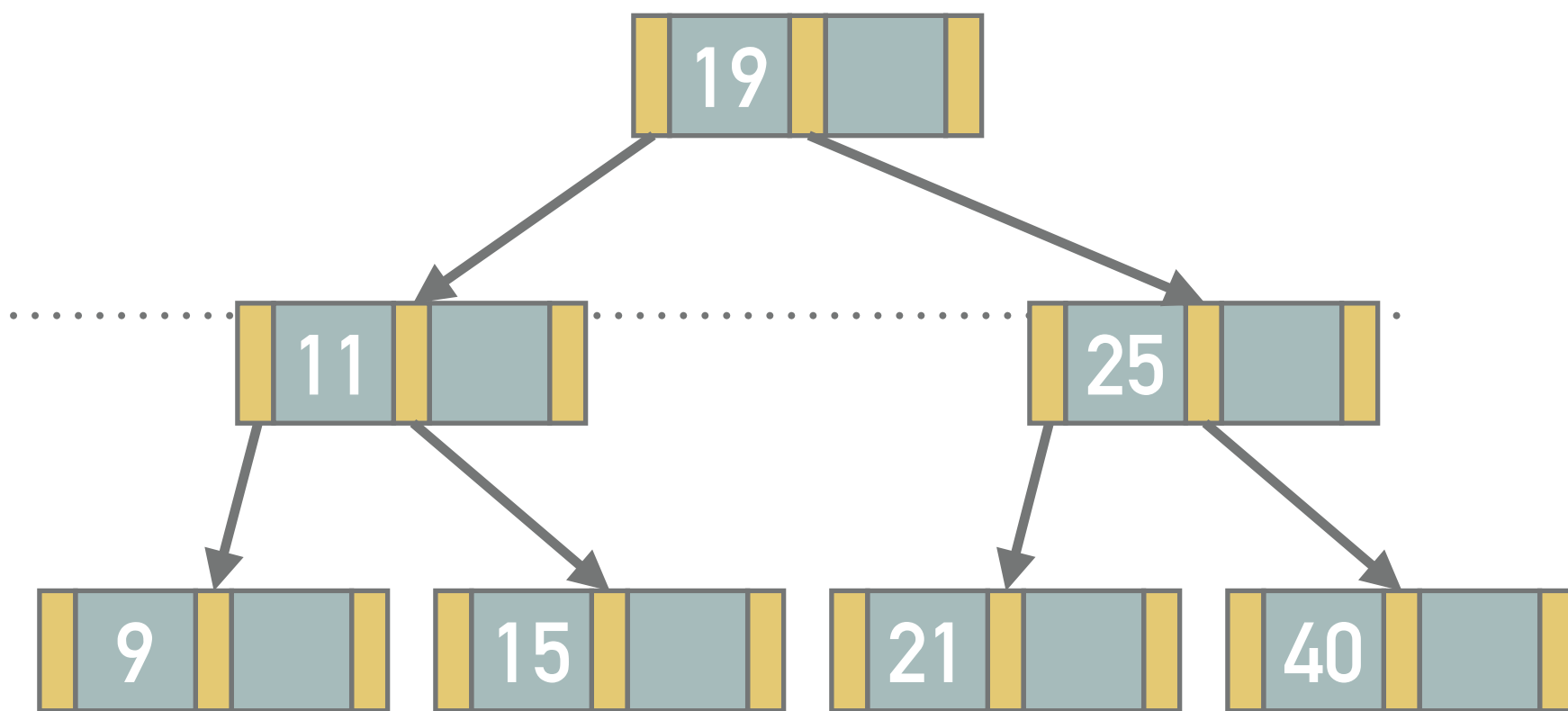
- Suppose a non-redundant B-tree of degree $m = 3$ (see the figure)
- First, illustrate the b-tree after insertion of records 11, 18 and 14
- Second, illustrate the b-tree after deletion of records 40 and 14



EXERCISE 1: SOLUTION 1/2

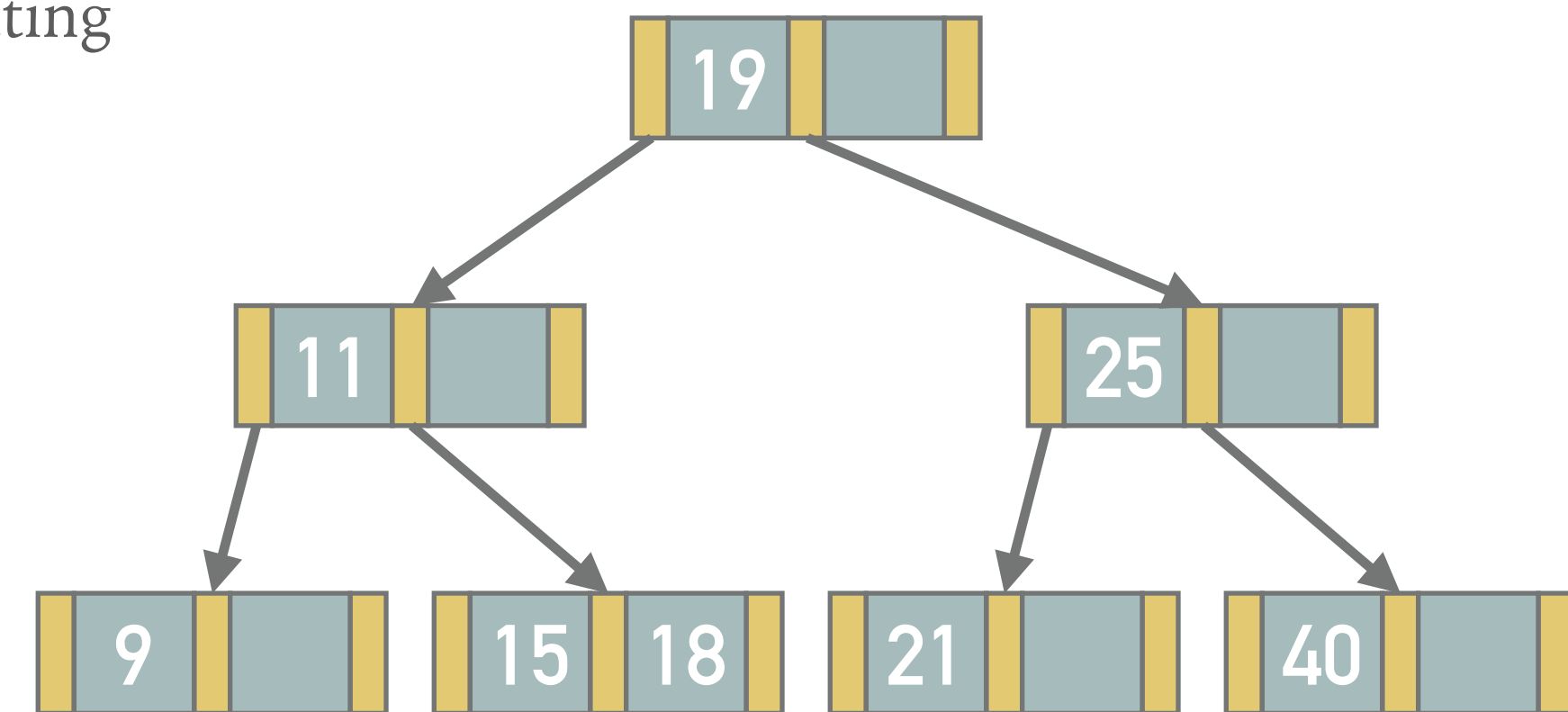
► Insertion of 11

- Key 11 is less than 19, i.e., we navigate to the left (leaf) node (9, 15)
- After the insertion, the leaf (9, 11, 15) contains too many keys, a splitting occurs
- Key 11 goes to the parent (11, 19, 25) that also contains too many keys, i.e., additional splitting occurs and the tree height is increased



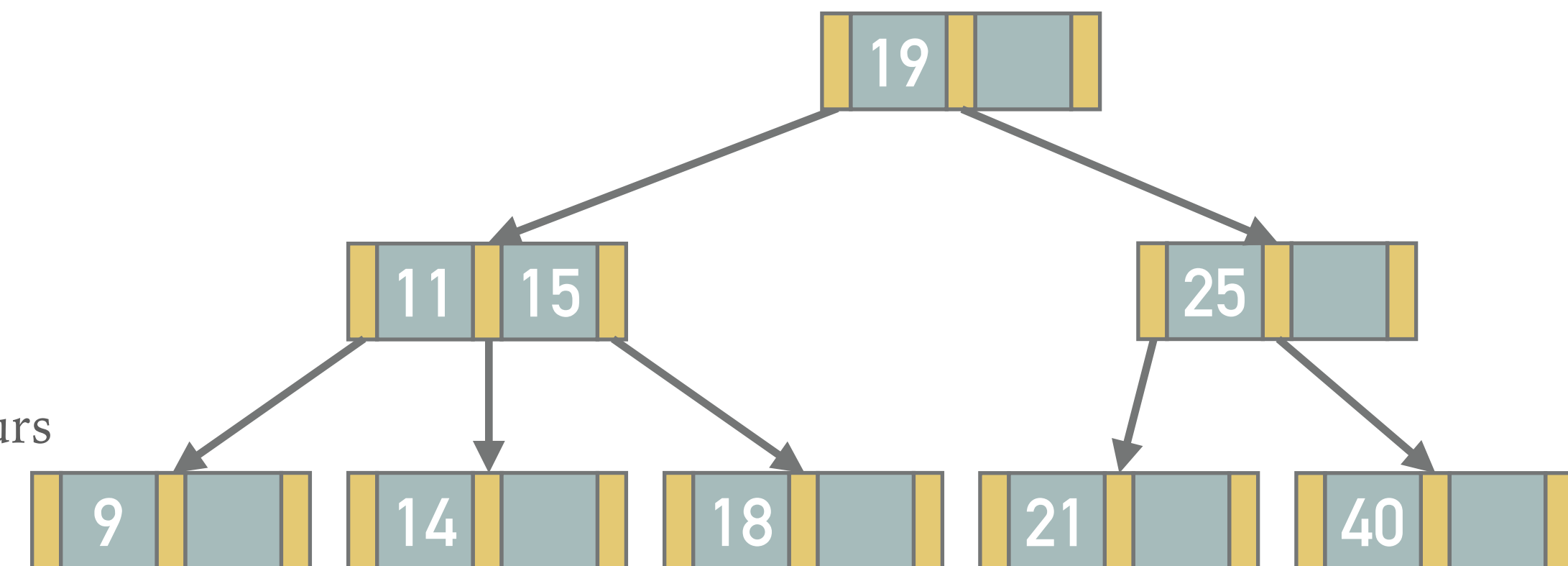
► Insertion of 18

- Key 18 is less than 19, i.e., we navigate to the left node (11)
- Key 18 is greater than 11, i.e., we navigate to the right node (15)
- Key 18 goes to the leaf (15,18)



► Insertion of 14

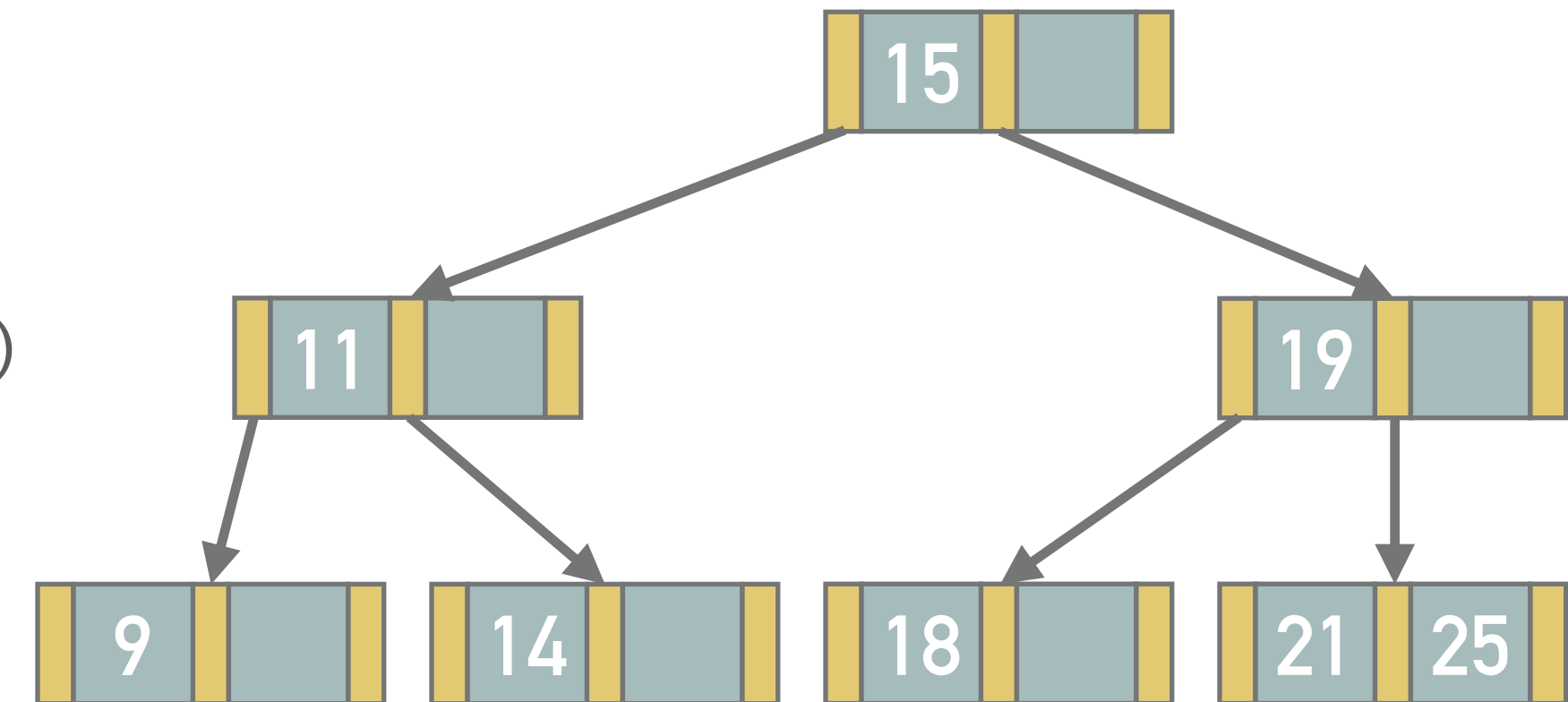
- Key 14 is less than 19, i.e., we navigate to the left node (11)
- Key 14 is greater than 11, i.e., we navigate to the right (leaf) node (15, 18)
- After the insertion, the leaf (14, 15, 18) contains too many keys, a splitting occurs
- Key 15 goes to the parent (11, 15)



EXERCISE 1: SOLUTION 2/2

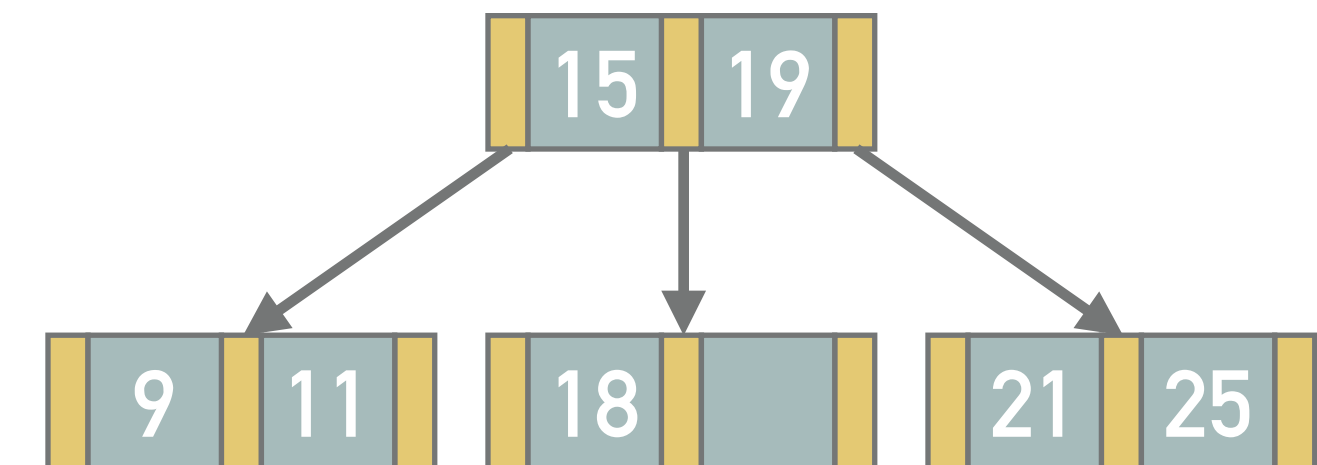
➤ Deletion of 40

- Key 40 is greater than 19, i.e., we navigate to the right node (25)
- Key 40 is greater than 25, i.e., we navigate to the right (leaf) node (40)
- Key 40 is removed, but we have to merge nodes (empty) and (25)
- Parent node (empty) has too few keys, therefore we take key from the left sibling (19)
- Finally, taking a key from the left sibling (11, 15), 15 goes to the root node



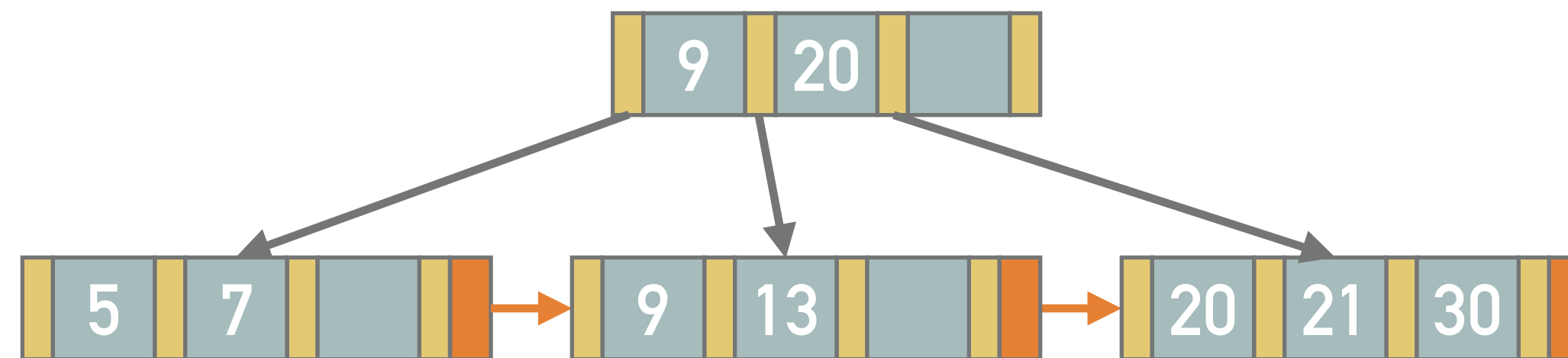
➤ Deletion of 14

- Key 14 is less than 15, i.e., we navigate to the left node (11)
- Key 14 is greater than 11, i.e., we navigate to the right (leaf) node (14)
- Key 14 is removed, causing the splitting of nodes (9) and (11)
- Finally, we have to merge nodes (15, 19) to establish a new root node



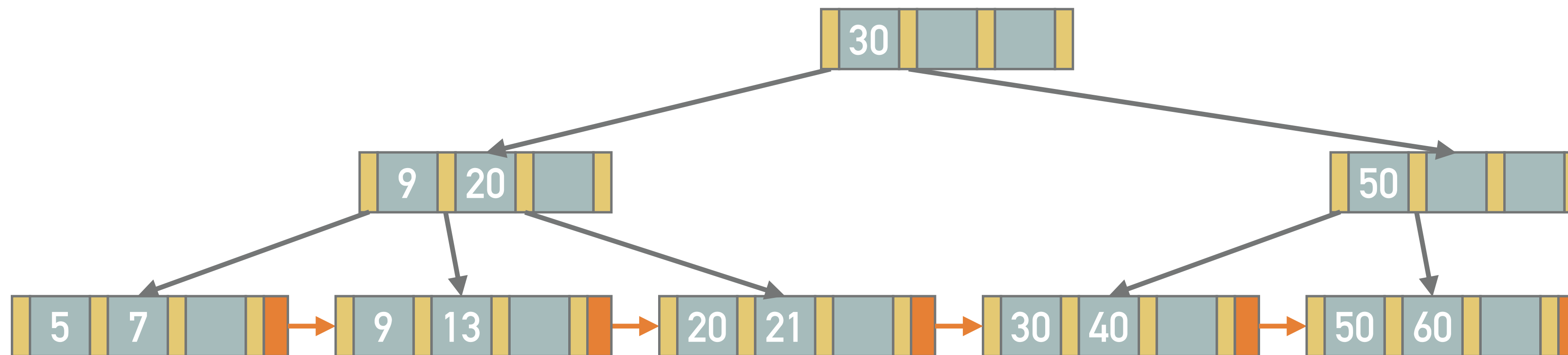
EXERCISE 2

- Suppose a B⁺-tree of degree $m = 4$ (see the figure)
 - Minimum number of children in a node is 2
- Illustrate the B⁺-tree after the insertion of records 40, 50 and 60



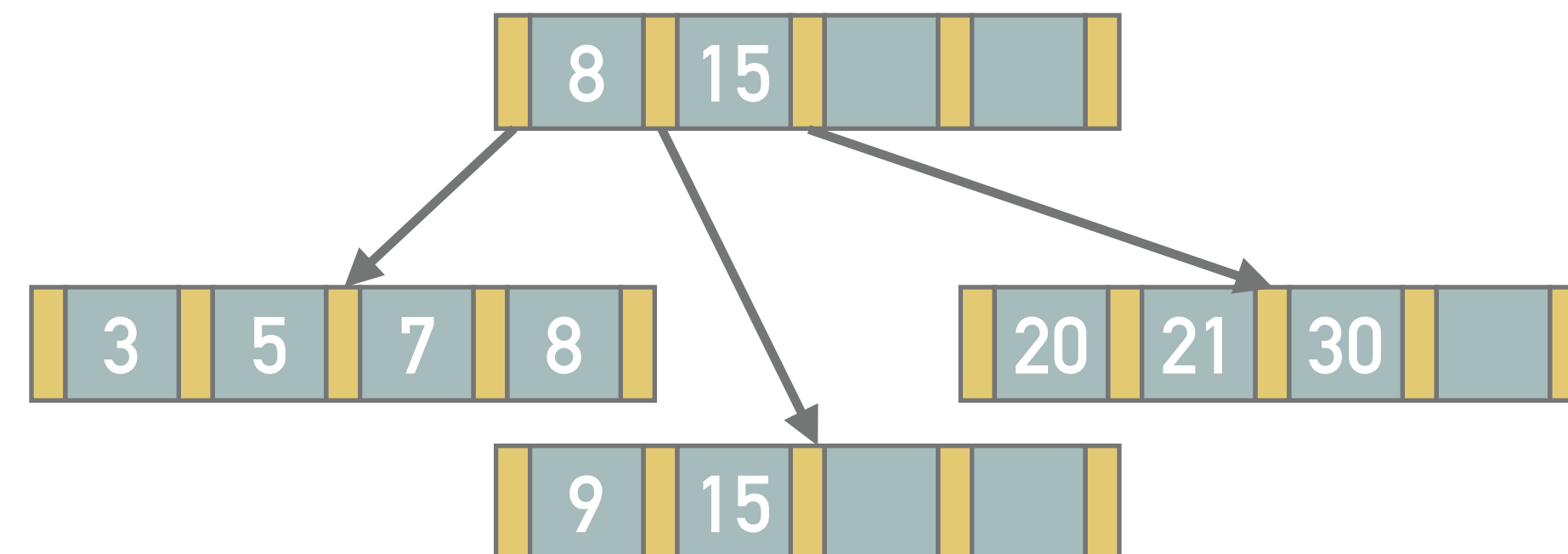
EXERCISE 2: SOLUTION

- The insertion of a record with key 40 splits the right leaf node into nodes (20, 21) and (30, 40)
 - The separating value 30 is inserted into the parent node where there is enough space
- The insertion of a key 50 is trivial, it goes to the node (30,40,50)
- The insertion of a key 60 splits the right leaf node into nodes (30,40) and (50,60)
 - The separating value 50 is inserted into the parent node (9,20,30,50) where it leads to the *cascade split*
 - The parent is split to nodes (9,20) and (50) while (30) goes to the root node



EXERCISE 3

- Continue with previous example and delete the records with keys 15, 9 and 8 from redundant B*-tree
- Finally, remove (single) additional key of your choice from the B*-tree
 - Illustrate and comment the removals step by step



EXERCISE 3: SOLUTION

- We can delete the record 15 by moving the record 20
- The record with key 9 will be removed while moving the record with key 8 and modification of the split value
- The record with key 8 will be removed while moving the record with key 7 and modification of the split value
- Removing another record would lead to merging of the three nodes into two

