BLG 335E – Analysis of Algorithms I
Fall 2017, Recitation 5
28.11.2017

R.A. Doğan Altan
daltan@itu.edu.tr – 4316
R.A. Cumali Türkmenoğlu
turkmenogluc@itu.edu.tr

Question 1

- Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length $m = 11$ using open addressing with the primary hash function $h'(k) = k \bmod m$.
- Illustrate the result of inserting these keys using **linear probing**, using **quadratic probing** with $c_1 = 1$ and $c_2 = 3$, and using **double hashing** with $h_2(k) = 1 + (k \bmod (m - 1))$.

Answer 1: Using Linear Probing

- **Linear probing:** $h(k,i) = (h'(k) + i) mod\ m$
- $h'(k) = k\ mod\ m$
- $m = 11,\ i = \{0, 1, 2, \dots, m-1\}$
- Set of keys: $\{10, 22, 31, 4, 15, 28, 17, 88, 59\}$

$h(10,0) = 10$
$h(22,0) = 0$
$h(31,0) = 9$
$h(4,0) = 4$
$h(15,0) = 4$
$\boldsymbol{h(15,1) = 5}$

$h(28,0) = 6$
$h(17,0) = 6$
$\boldsymbol{h(17,1) = 7}$
$h(88,0) = 0$
$\boldsymbol{h(88,1) = 1}$

$h(59,0) = 4$
$\boldsymbol{h(59,1) = 5}$
$\boldsymbol{h(59,2) = 6}$
$\boldsymbol{h(59,3) = 7}$
$\boldsymbol{h(59,4) = 8}$

- The resulting hash table:
  - $H = \{22, 88, nil, nil, 4, 15, 28, 17, 59, 31, 10\}$

Answer 1: Using Quadratic Probing

- **Quadratic probing:** $h(k,i) = (h'(k) + c_1 i + c_2 i^2) mod\ m$
- $h'(k) = k\ mod\ m, c_1 = 1, c_2 = 3$
- $m = 11,\ i = \{0, 1, 2, \dots, m-1\}$
- Set of keys: $\{10, 22, 31, 4, 15, 28, 17, 88, 59\}$

$h(10,0) = 10$	$h(17,0) = 6$	$\boldsymbol{h(88,3) = 8}$
$h(22,0) = 0$	$\boldsymbol{h(17,1) = 10}$	$\boldsymbol{h(88,4) = 8}$
$h(31,0) = 9$	$\boldsymbol{h(17,2) = 9}$	$\boldsymbol{h(88,5) = 3}$
$h(4,0) = 4$	$\boldsymbol{h(17,3) = 3}$	$\boldsymbol{h(88,6) = 4}$
$h(15,0) = 4$	$h(88,0) = 0$	$\boldsymbol{h(88,7) = 0}$
$\boldsymbol{h(15,1) = 8}$	$\boldsymbol{h(88,1) = 4}$	$\boldsymbol{h(88,8) = 2}$
$h(28,0) = 6$	$\boldsymbol{h(88,2) = 3}$	$h(59,0) = 4$
		$\boldsymbol{h(59,1) = 8}$
		$\boldsymbol{h(59,2) = 7}$

- The resulting hash table:
  - $H = \{22, nil, 88, 17, 4, nil, 28, 59, 15, 31, 10\}$

Answer 1: Using Double Hashing

- **Double Hashing:** $h(k,i) = \big(h_1(k) + ih_2(k)\big) mod\ m$
- $h_1(k) = k\ mod\ m$ and $h_2(k) = 1 + k\ mod\ (m-1)$
- $m = 11,\ i = \{0, 1, 2, \dots, m-1\}$
- Set of keys: $\{10, 22, 31, 4, 15, 28, 17, 88, 59\}$

$h(10,0) = 10$	$\boldsymbol{h(15,2) = 5}$	$\boldsymbol{h(88,2) = 7}$
$h(22,0) = 0$	$h(28,0) = 6$	$h(59,0) = 4$
$h(31,0) = 9$	$h(17,0) = 6$	$\boldsymbol{h(59,1) = 3}$
$h(4,0) = 4$	$\boldsymbol{h(17,1) = 3}$	$\boldsymbol{h(59,2) = 2}$
$h(15,0) = 4$	$h(88,0) = 0$	
$\boldsymbol{h(15,1) = 10}$	$\boldsymbol{h(88,1) = 9}$	

- The resulting hash table:
  - $H = \{22, nil, 59, 17, 4, 15, 28, 88, nil, 31, 10\}$

Question 2

- Insert the following sequence of numbers into a 2-3-4 tree
  - {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}

Solution 2

- **2-3-4 tree:** Perfect balance by allowing 1, 2, or 3 keys per node:
  - 2-node: one key, two children.
  - 3-node: two keys, three children.
  - 4-node: three keys, four children.
- Every path from root to leaf has the same length.

Solution 2 (Cont.)

- {**53**, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}

Solution 2 (Cont.)

- {53, **27**, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}

27	53

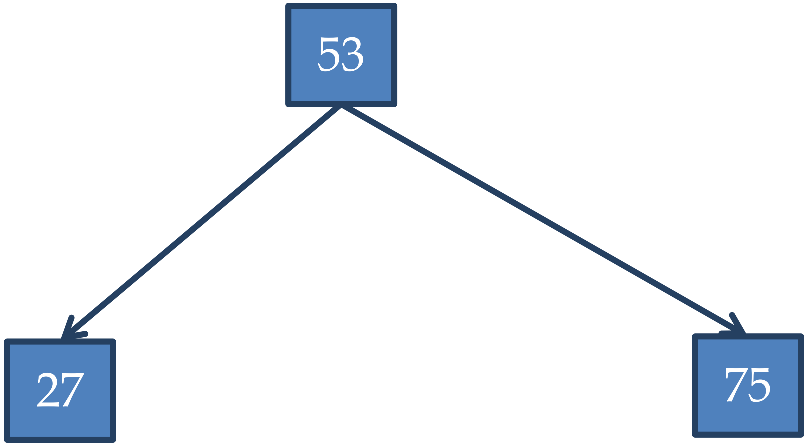
Solution 2 (Cont.)

- {53, 27, **75**, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}

27	53	75

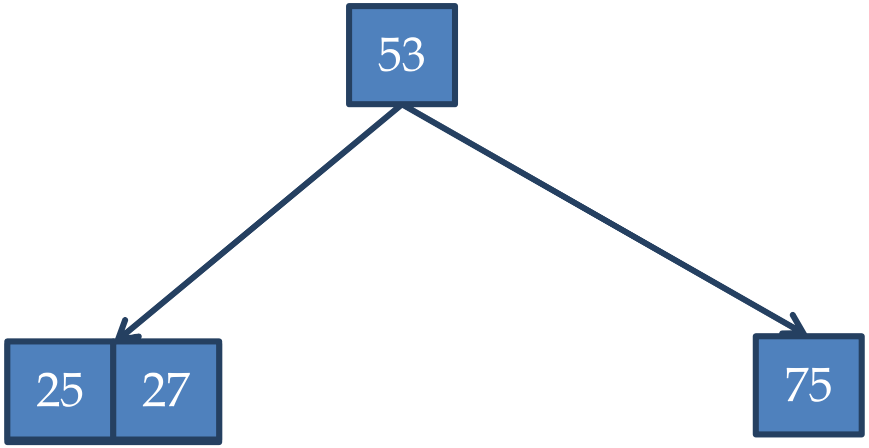
Solution 2 (Cont.)

- {53, 27, 75, **25**, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



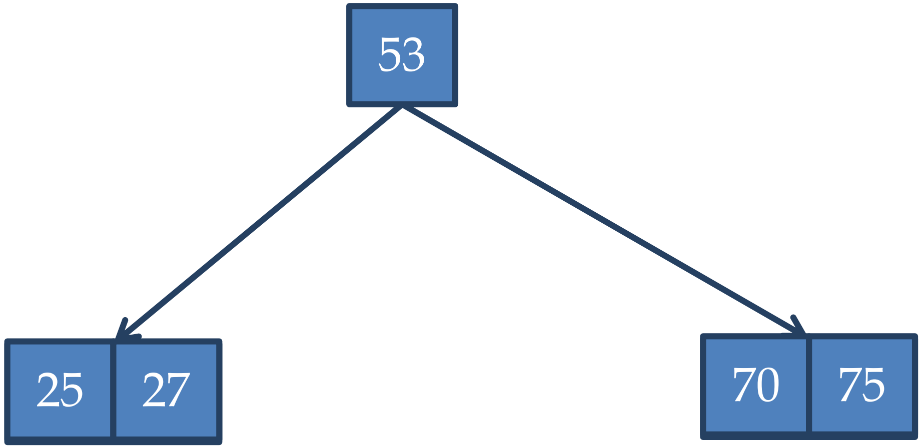
Solution 2 (Cont.)

- {53, 27, 75, **25**, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



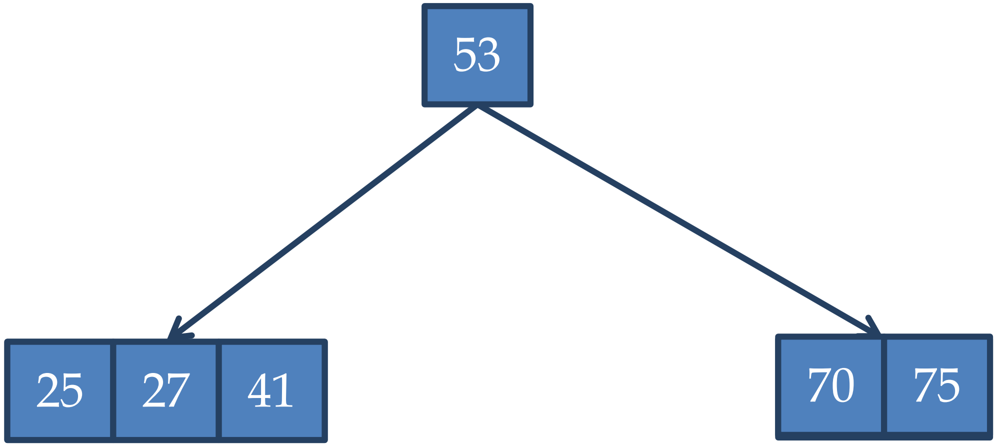
Solution 2 (Cont.)

- {53, 27, 75, 25, **70**, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}



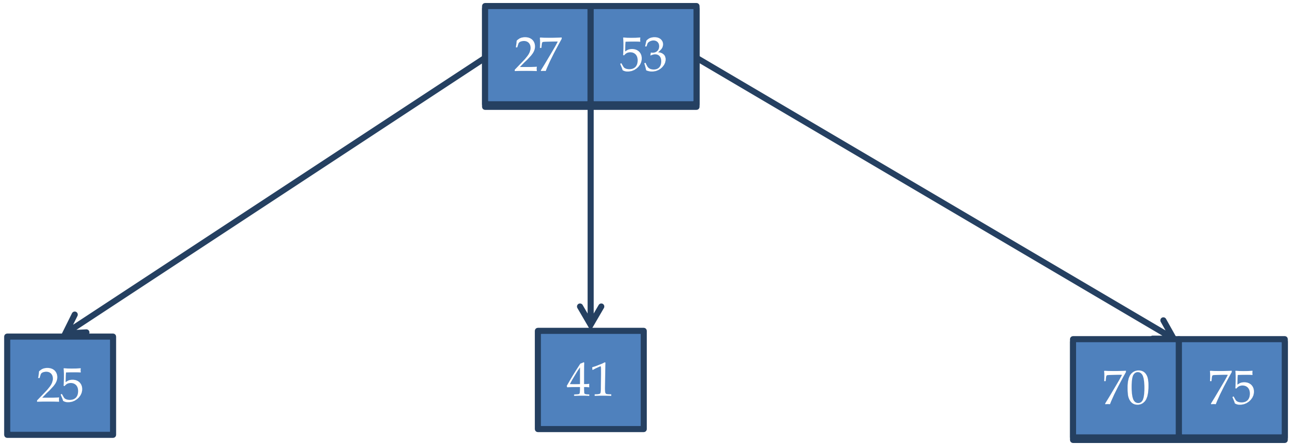
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, **41**, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}



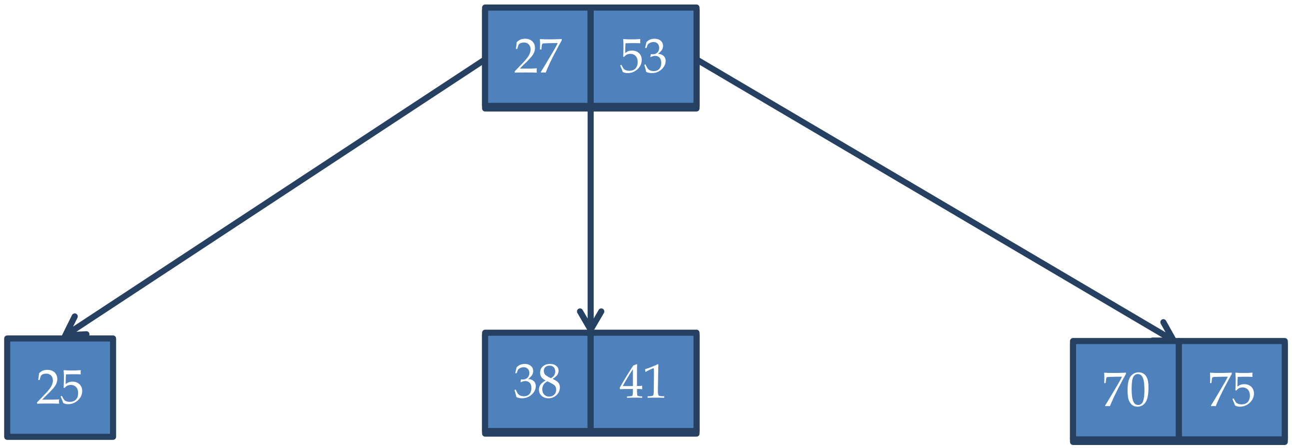
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, **38**, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



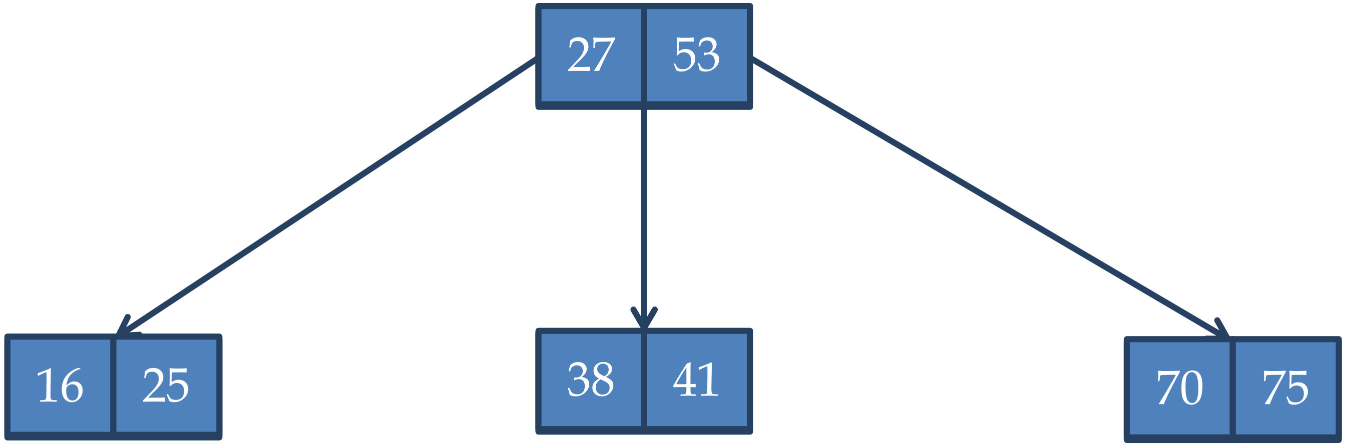
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, **38**, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



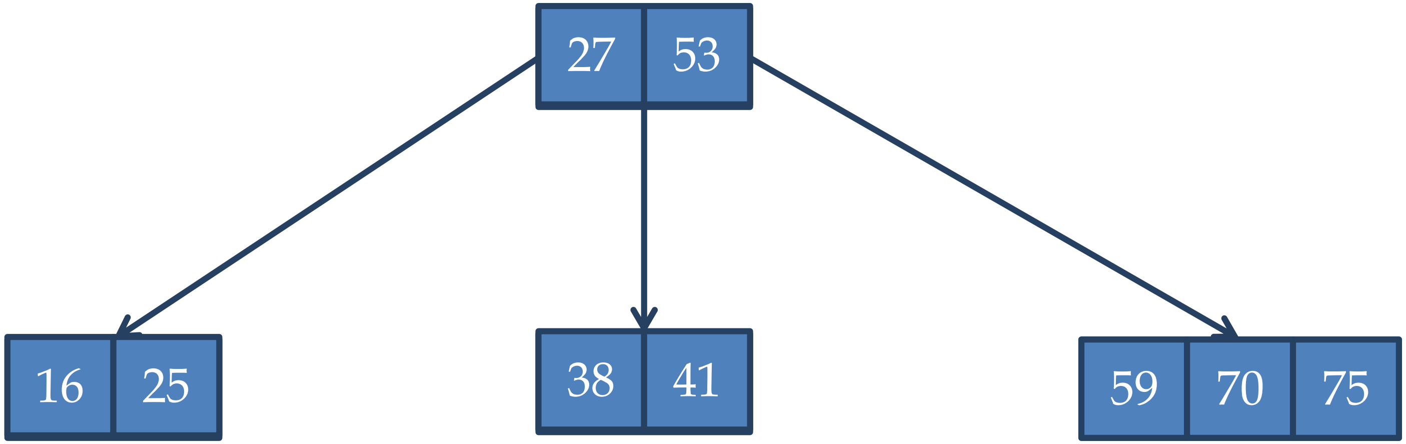
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, **16**, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}



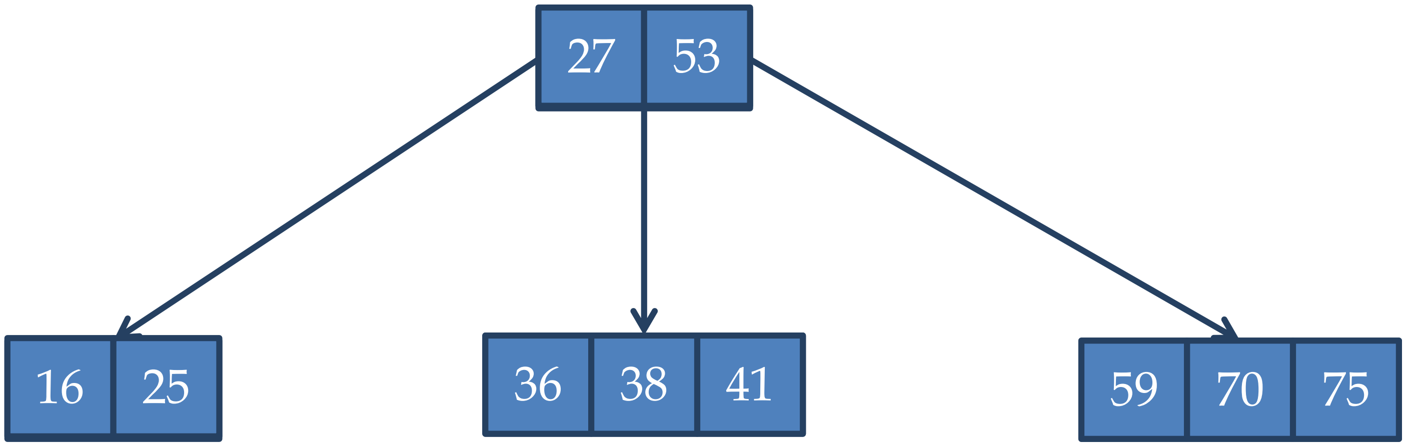
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, **59**, 36, 73, 65, 60, 46, 55, 33, 68, 79, 48}



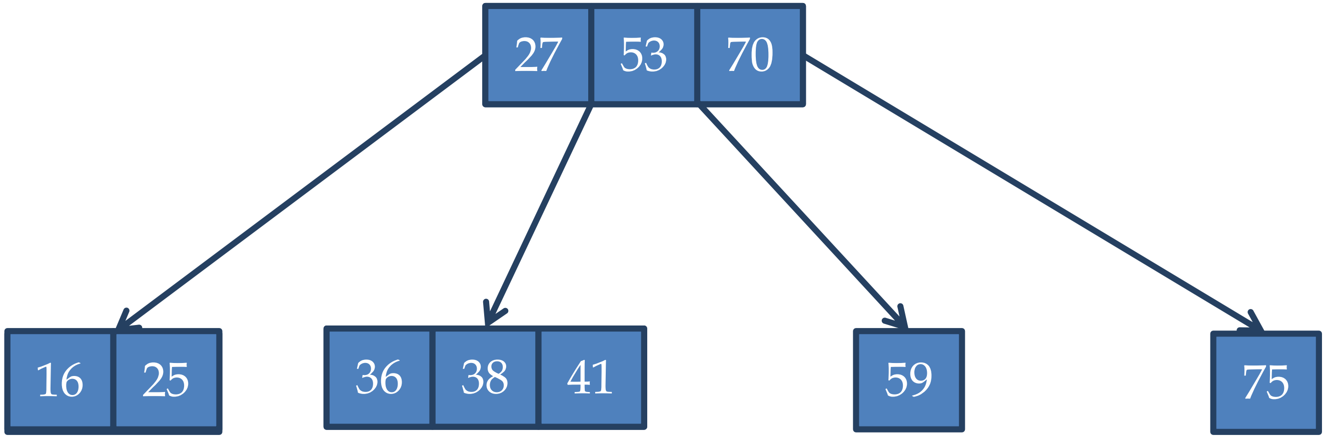
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, **36**, 73, 65, 60, 46, 55, 33, 68, 79, 48}



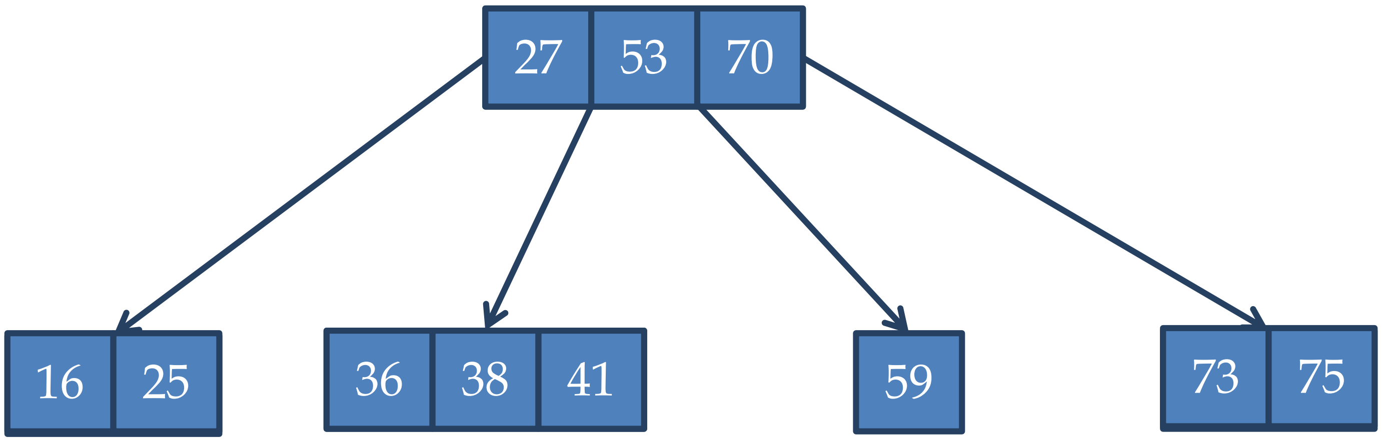
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, **73**, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



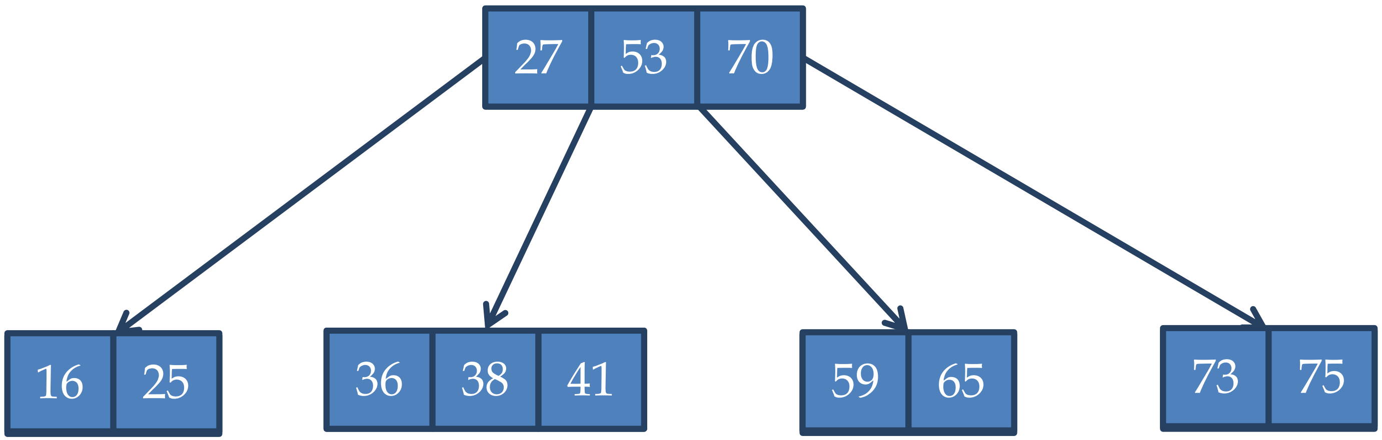
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, **73**, 65, 60, 46, 55, 33, 68, 79, 48} → causes a split



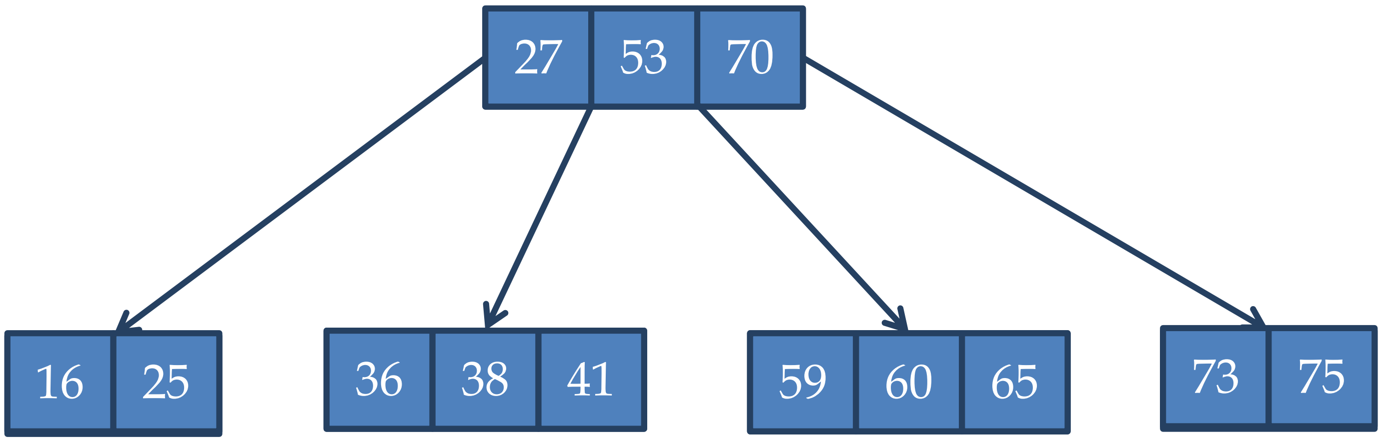
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, **65**, 60, 46, 55, 33, 68, 79, 48}



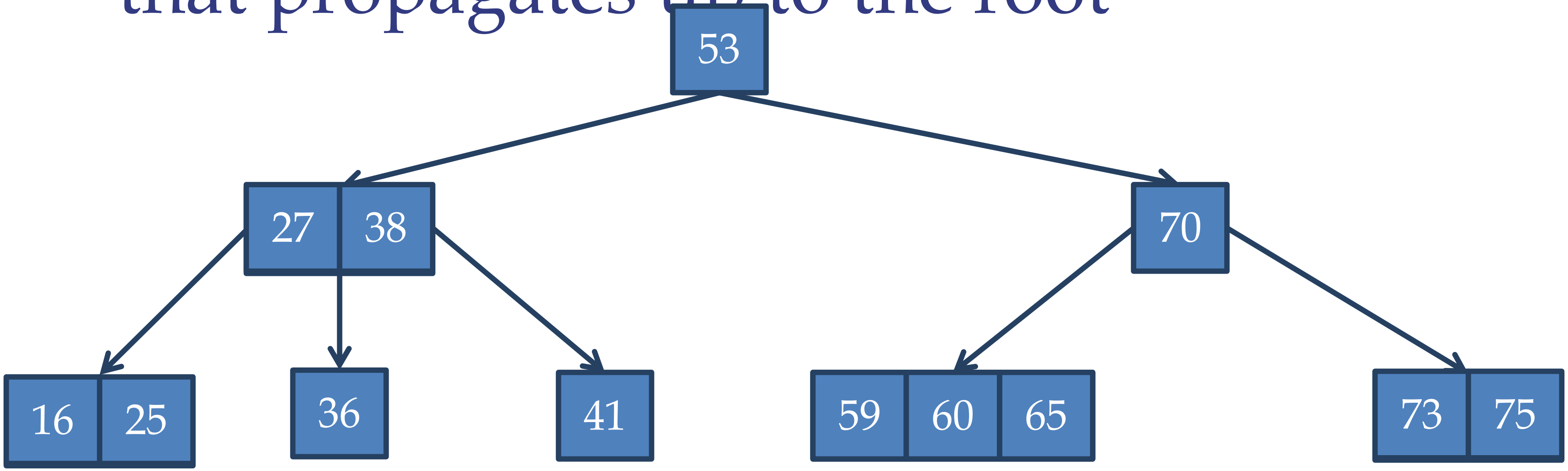
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, **60**, 46, 55, 33, 68, 79, 48}



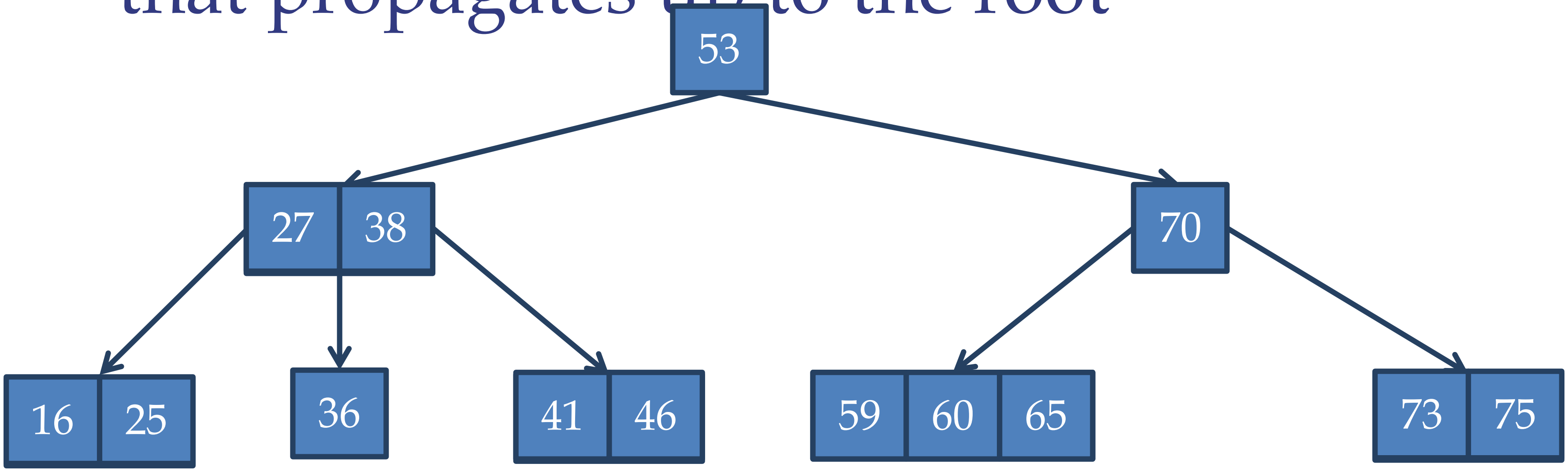
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, **46**, 55, 33, 68, 79, 48} → causes a split that propagates up to the root



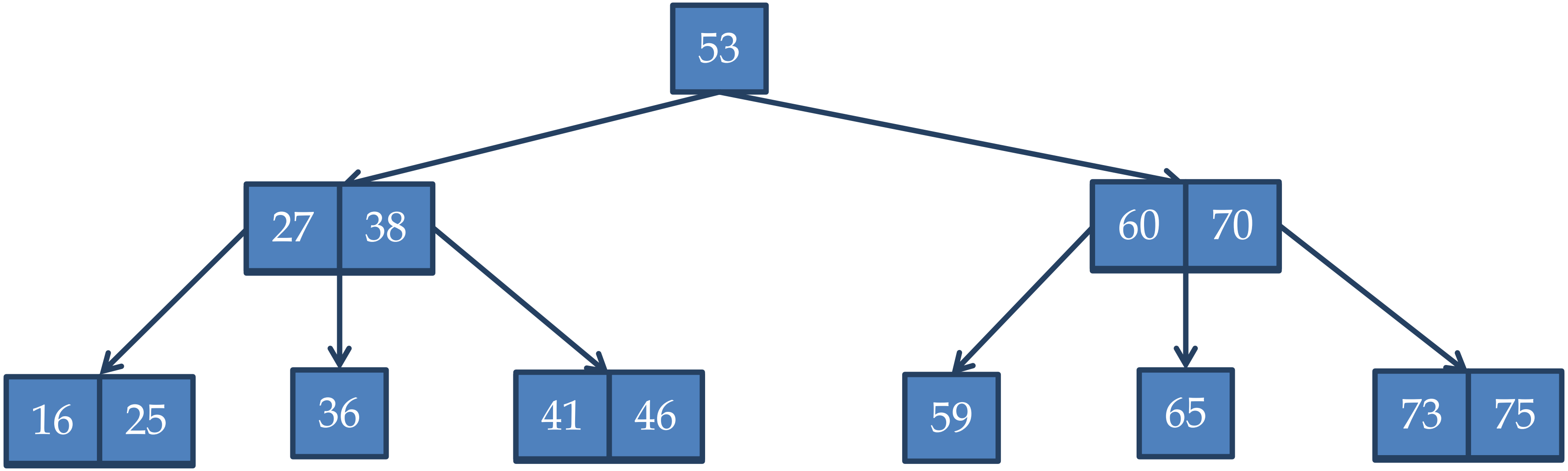
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, **46**, 55, 33, 68, 79, 48} → causes a split that propagates up to the root



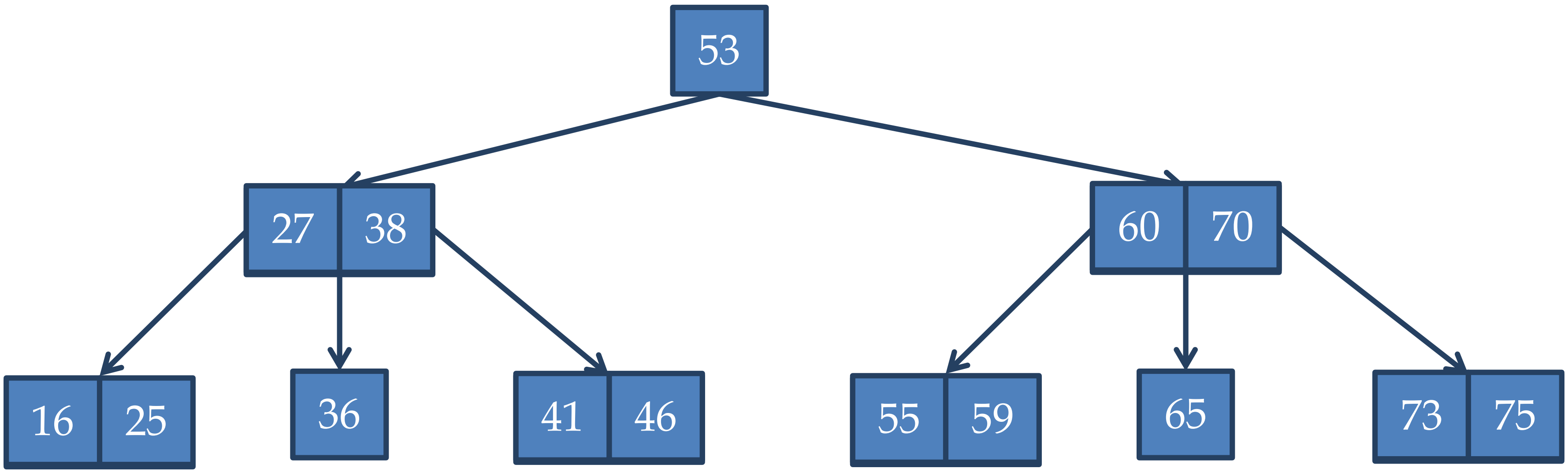
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, **55**, 33, 68, 79, 48} → causes a split



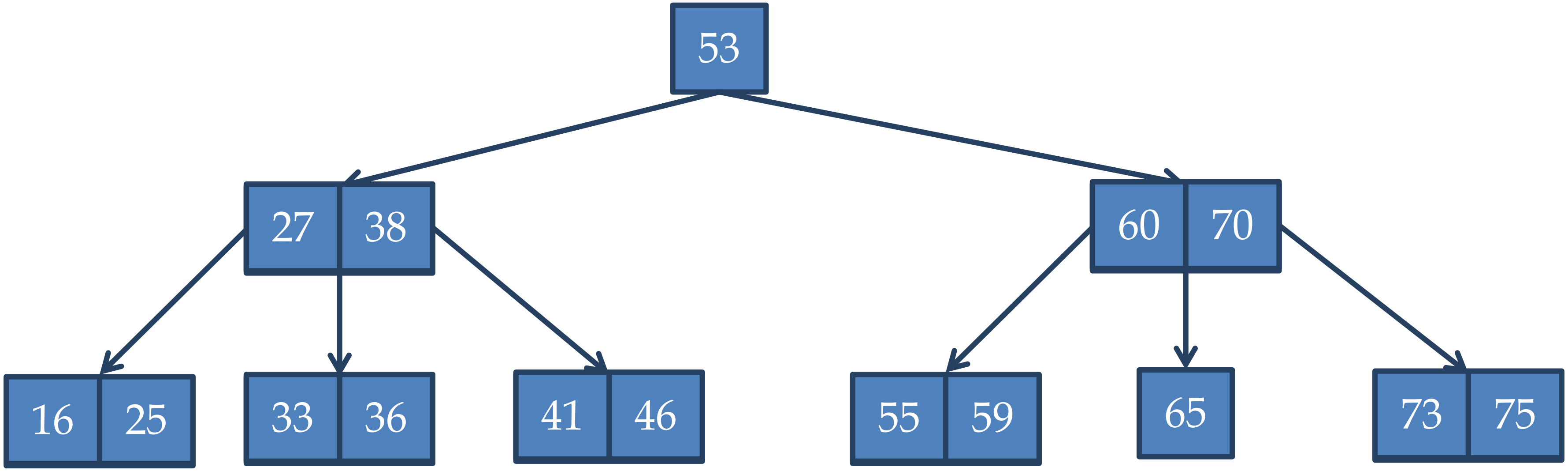
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, **55**, 33, 68, 79, 48} → causes a split



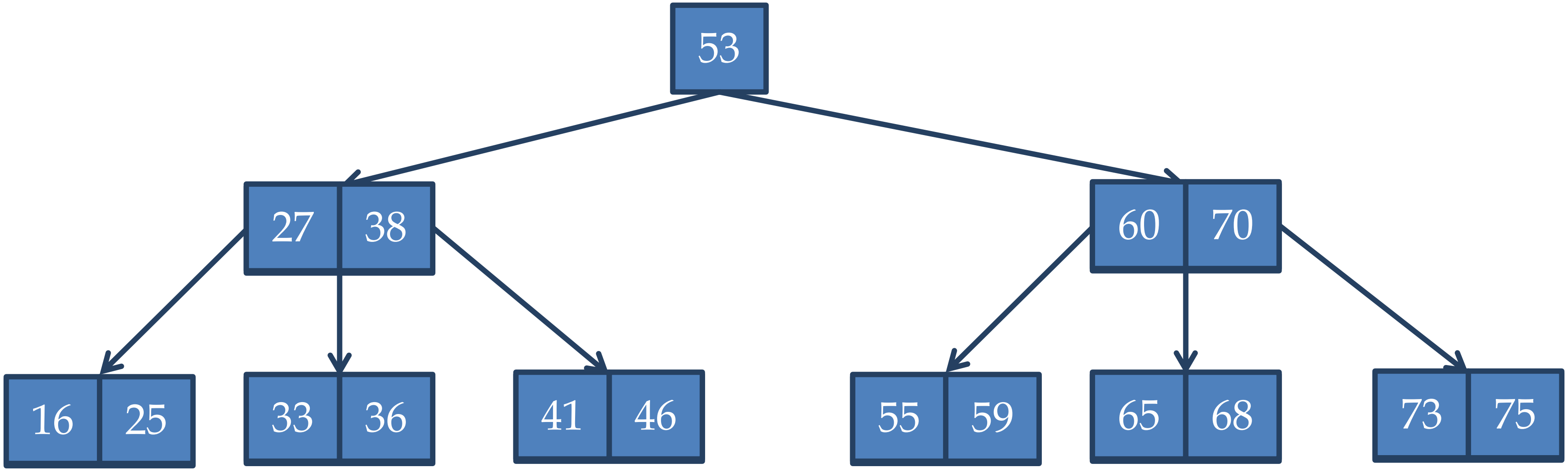
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, **33**, 68, 79, 48}



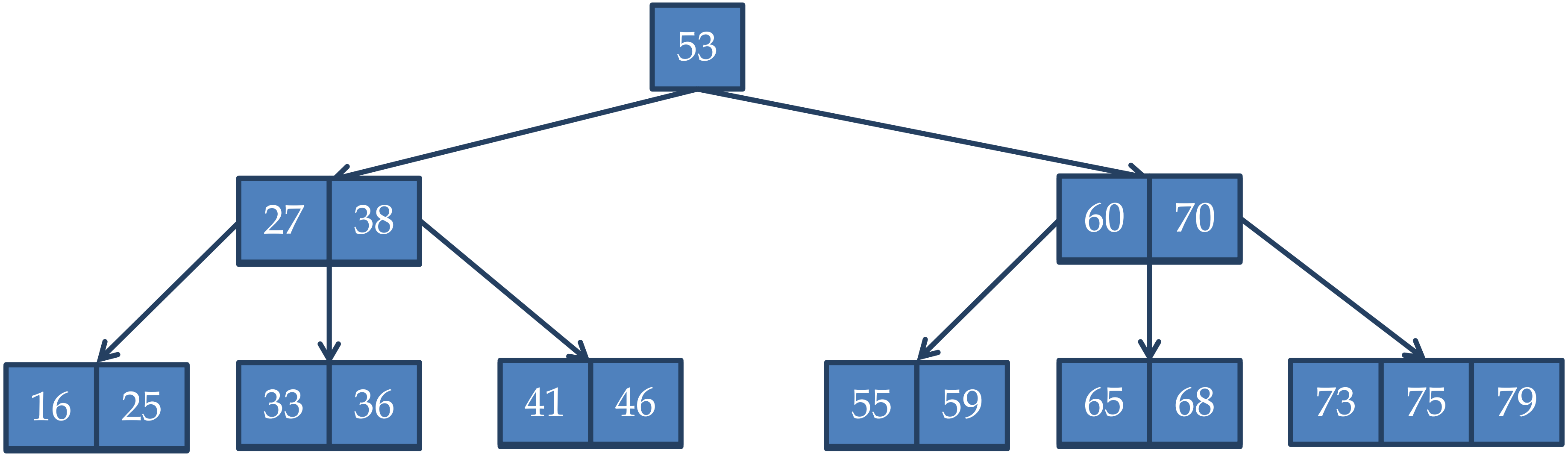
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, **68**, 79, 48}



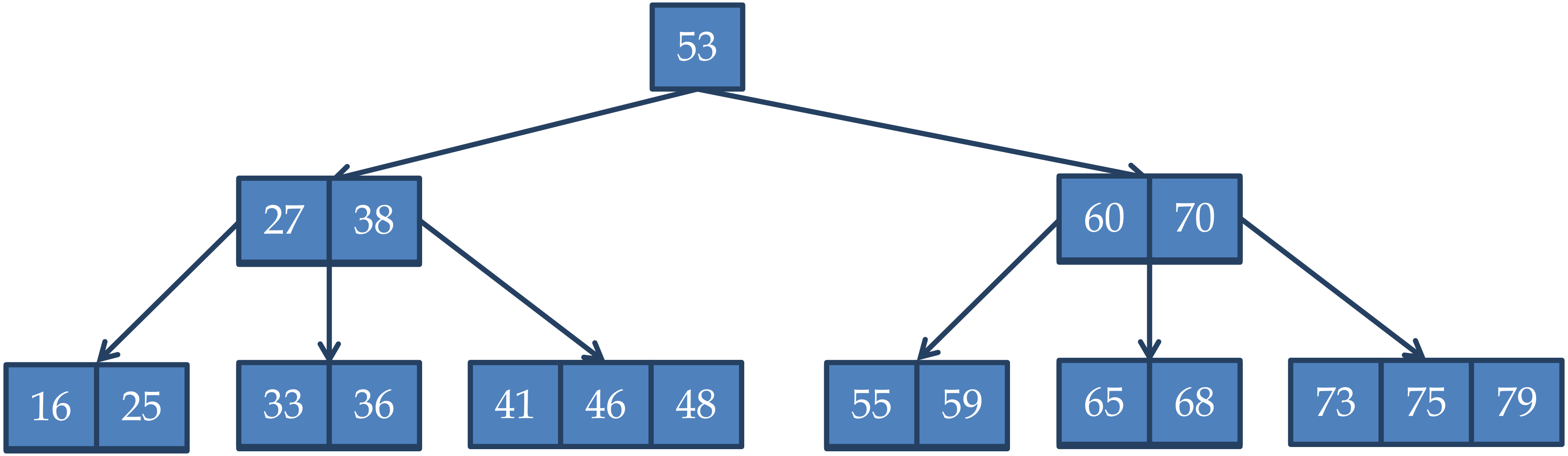
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, **79**, 48}



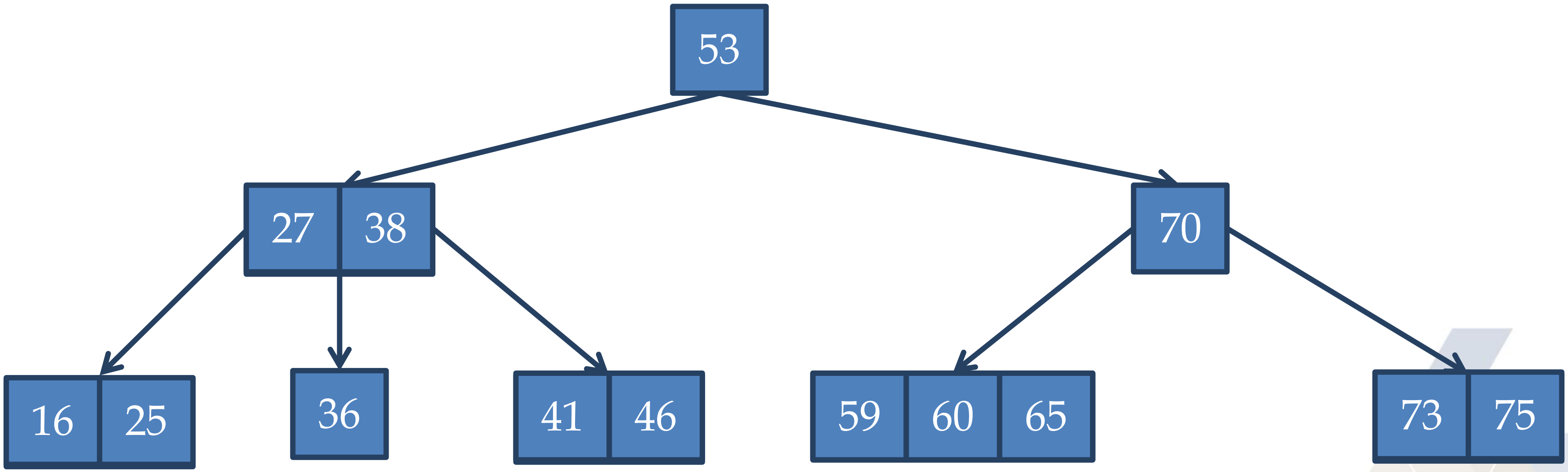
Solution 2 (Cont.)

- {53, 27, 75, 25, 70, 41, 38, 16, 59, 36, 73, 65, 60, 46, 55, 33, 68, 79, **48**}



Question 3

- 2-3-4 trees are balanced and can be searched in $O(logn)$, but they have different node structures.
- To get 2-3-4 tree advantages in a binary tree format, we can represent it as a red-black tree.
- Convert the following 2-3-4 tree to a red-black tree

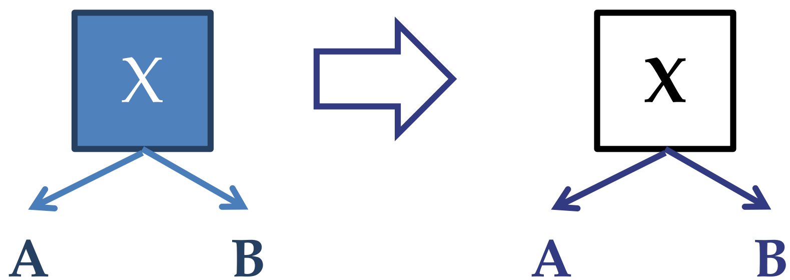


Solution 3

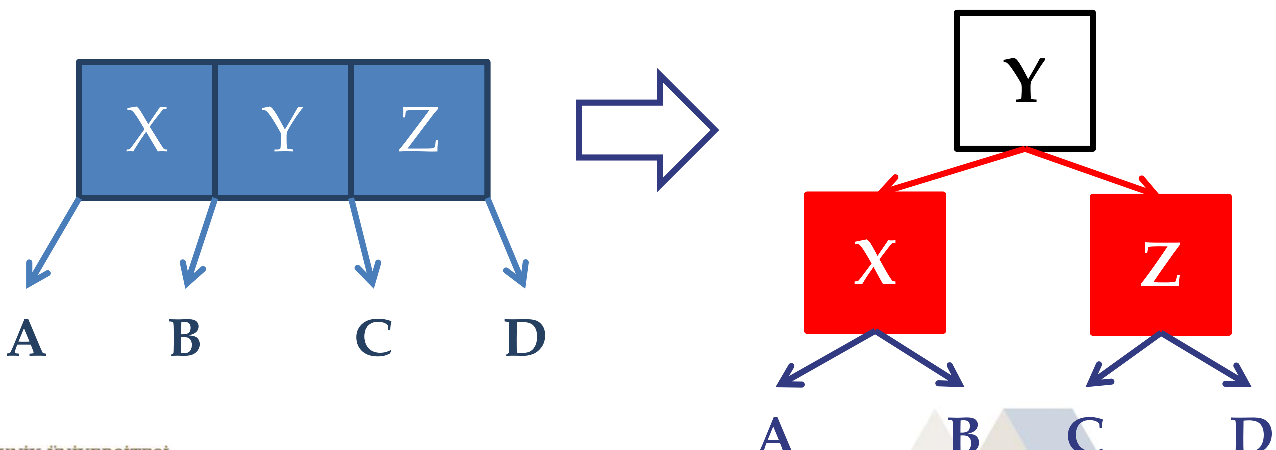
- **Properties of a red-black tree:**
  - the root is always black
  - **black condition**: every path from the root to a leaf node has the same number of black nodes
  - **red condition**: every red node has a black parent

Solution 3 (Cont.)

- **2-nodes:** can be represented with a **black** node

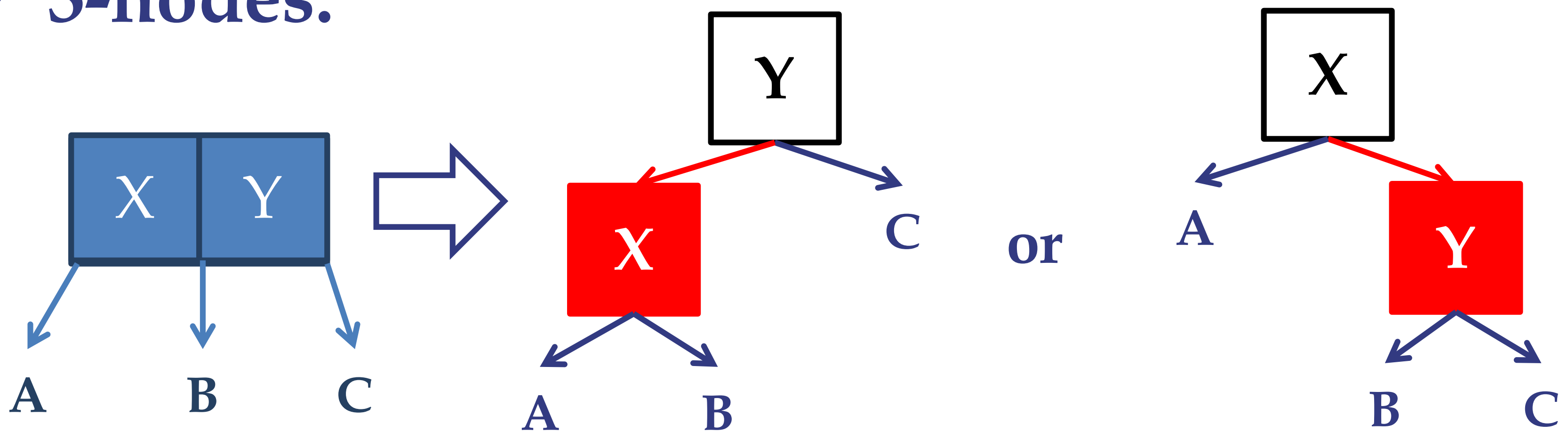


- **4-nodes:** center value becomes the parent (**black**) and the others become children (**red**)



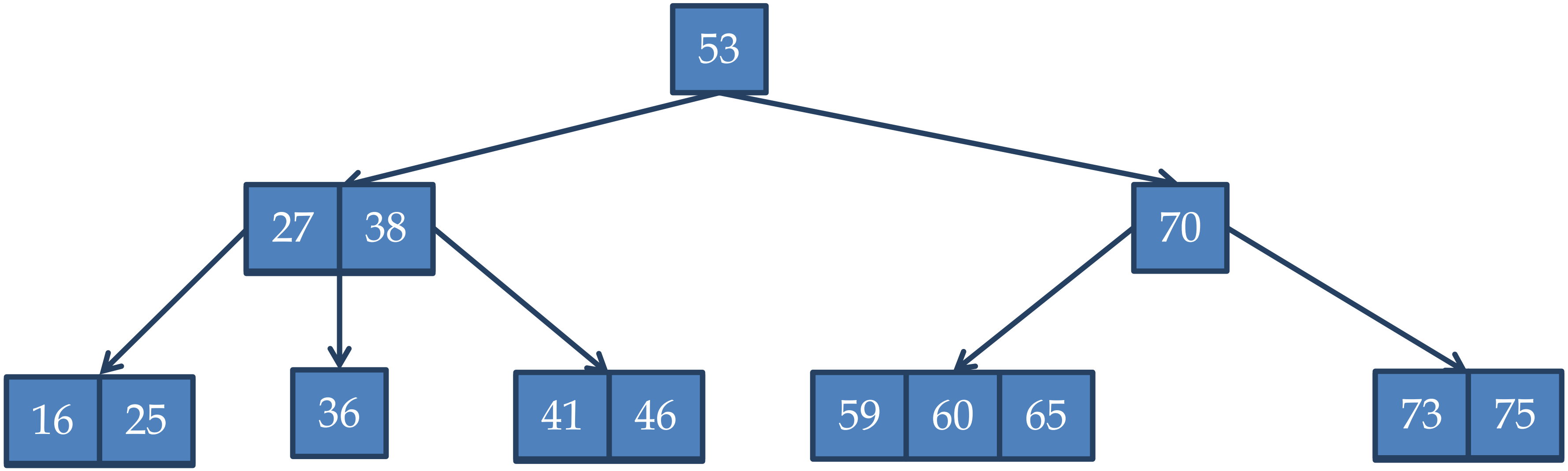
Solution 3 (Cont.)

- **3-nodes:**



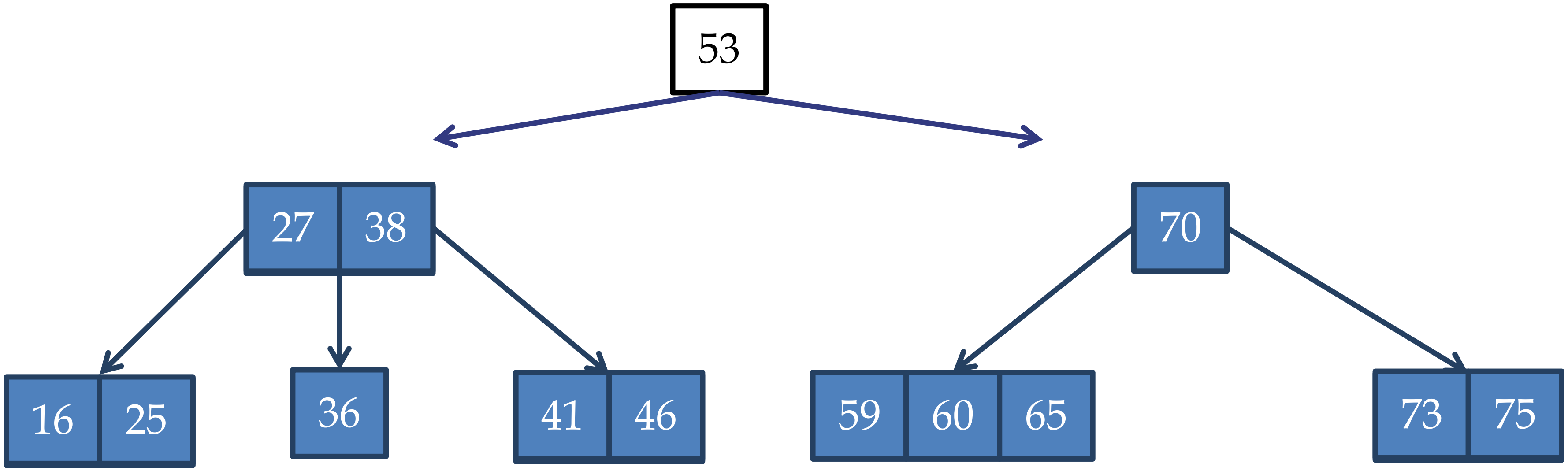
- **Note:**
  1. Red-black trees are not unique
  2. However, the corresponding 2-3-4 tree is unique

Solution 3 (Cont.)

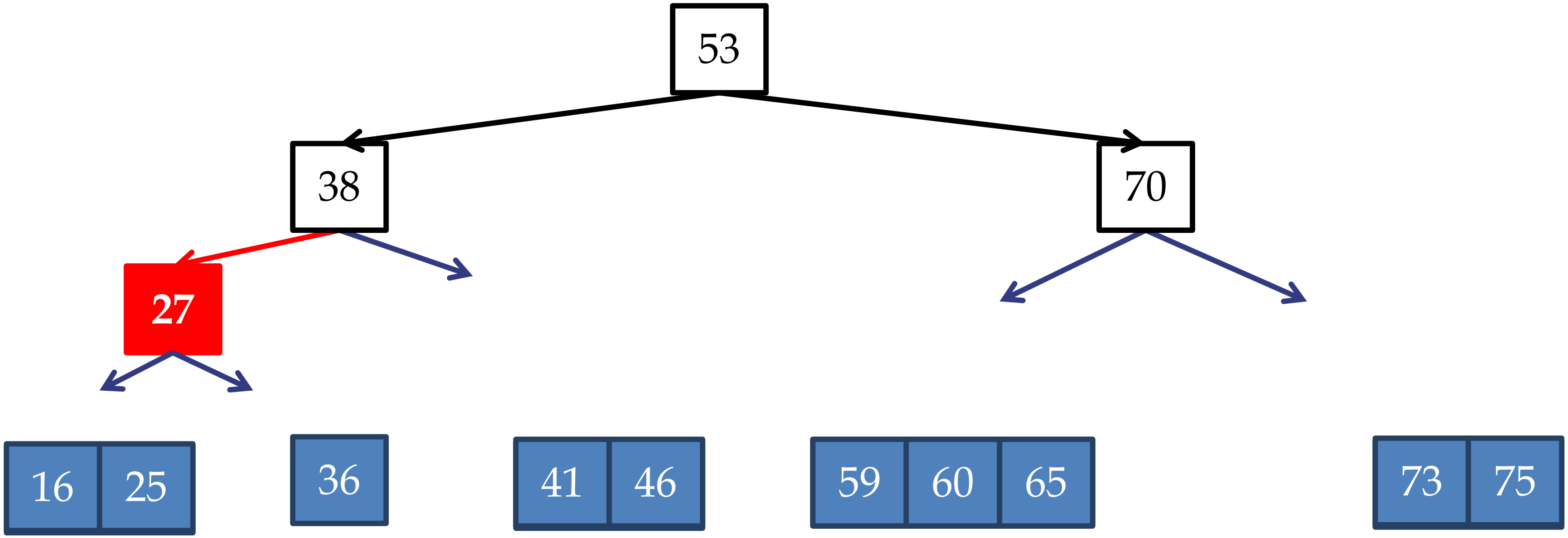


- Top-down conversion algorithm (start at the root):
  1. Apply red-black tree representation to each node
  2. Repeat for next level…

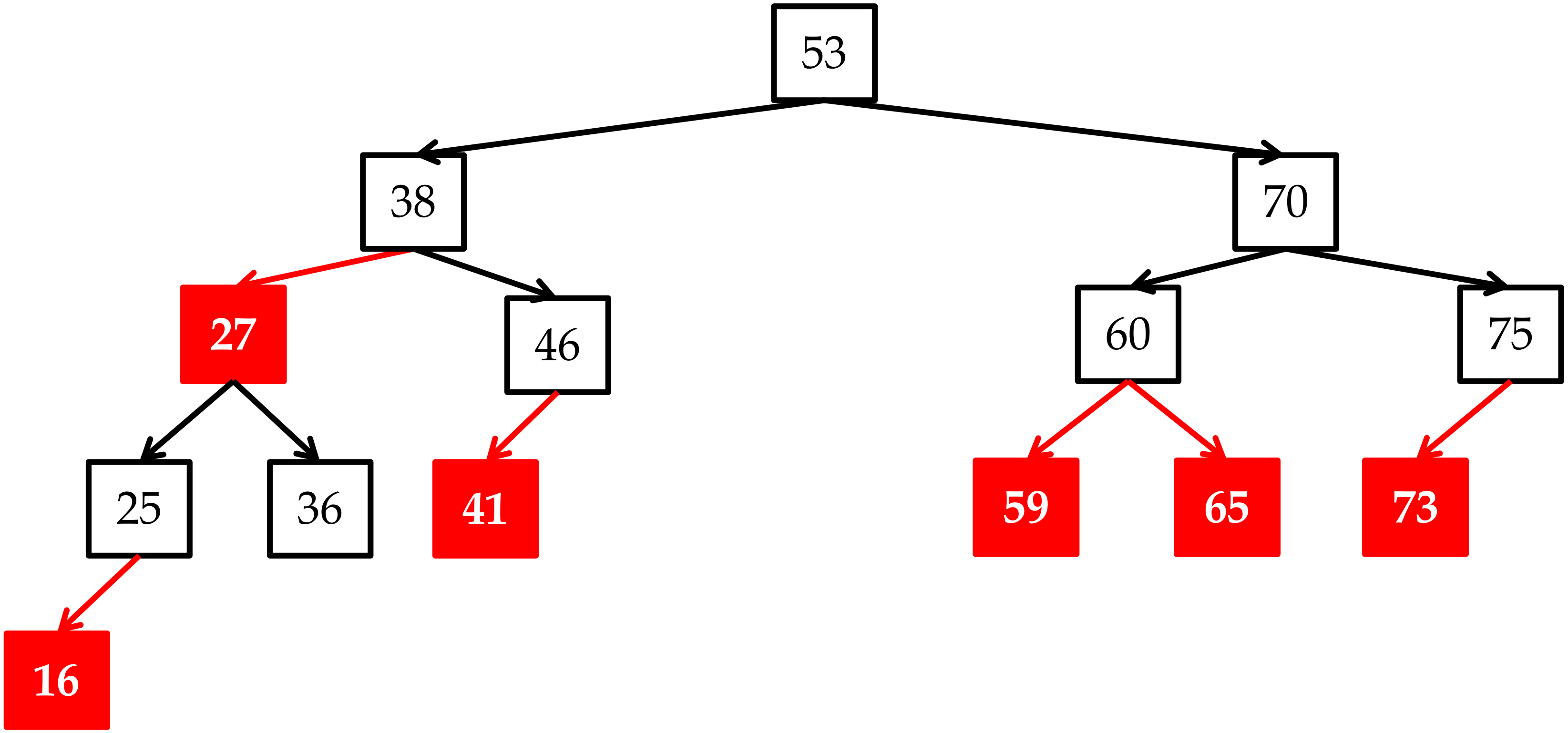
Solution 3 (Cont.)



Solution 3 (Cont.)



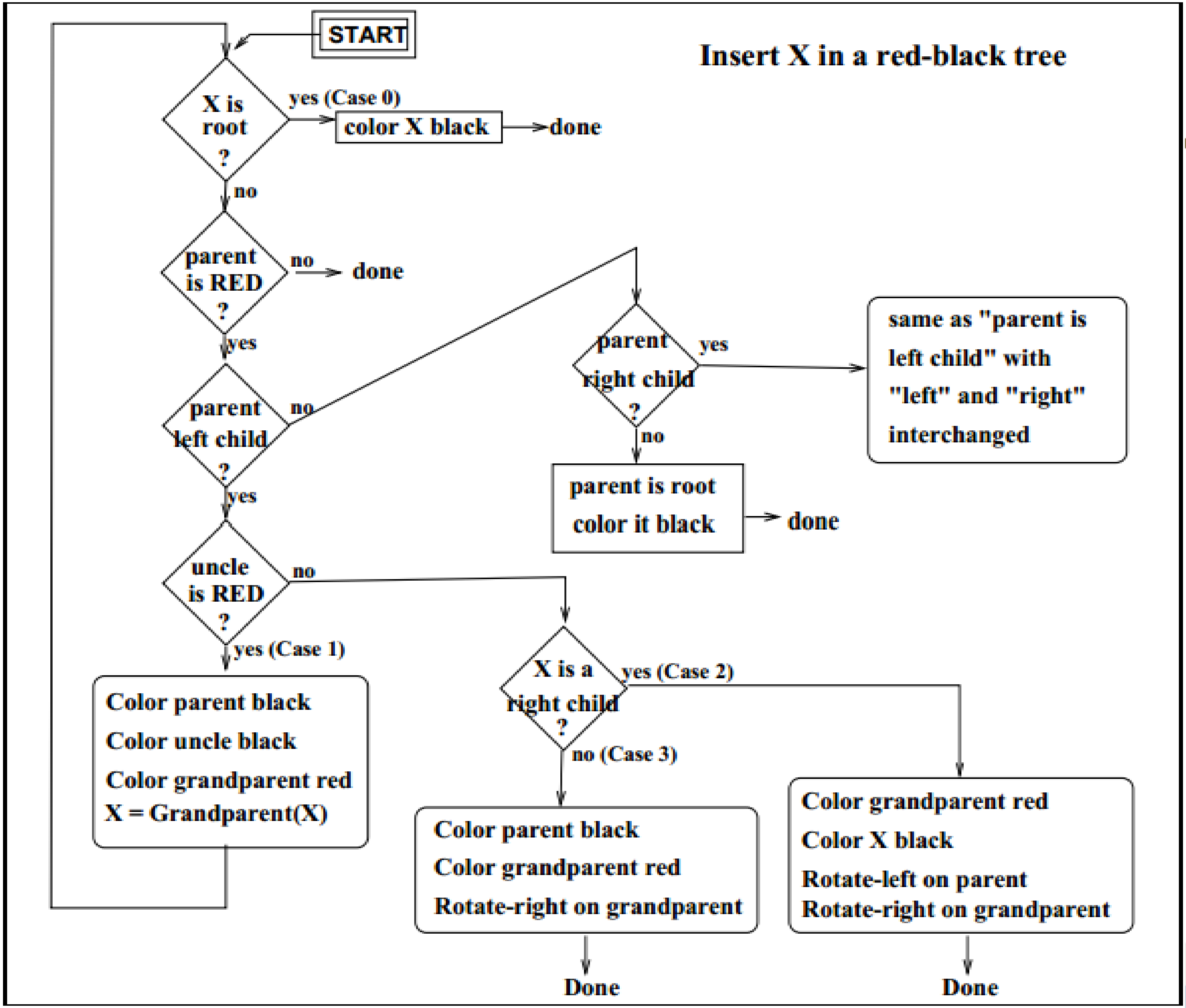
Solution 3 (Cont.)



Question 4

- Insert the following sequence of numbers into a red-black tree
  - {2, 1, 4, 5, 9, 3, 6, 7}

Insert X in a red-black tree



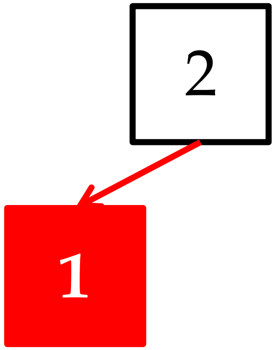
Solution 4

- {**2**, 1, 4, 5, 9, 3, 6, 7}

$\boxed{2}$

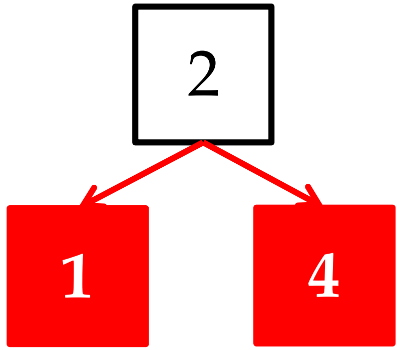
Solution 4 (Cont.)

- {2, **1**, 4, 5, 9, 3, 6, 7}



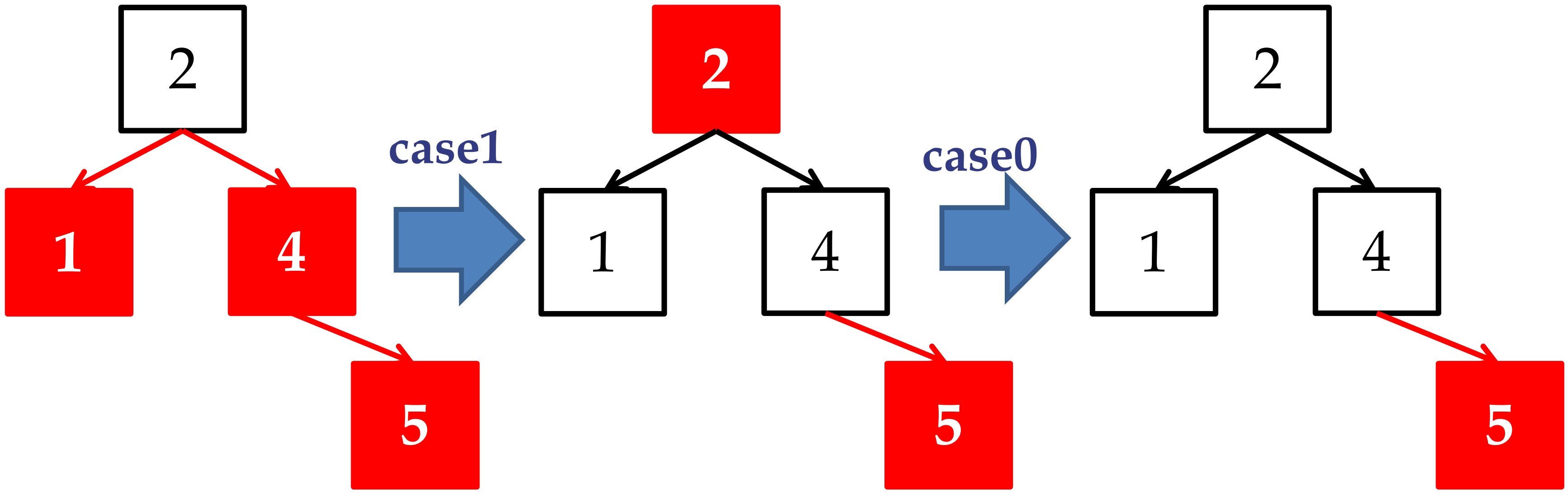
Solution 4 (Cont.)

- {2, 1, **4**, 5, 9, 3, 6, 7}



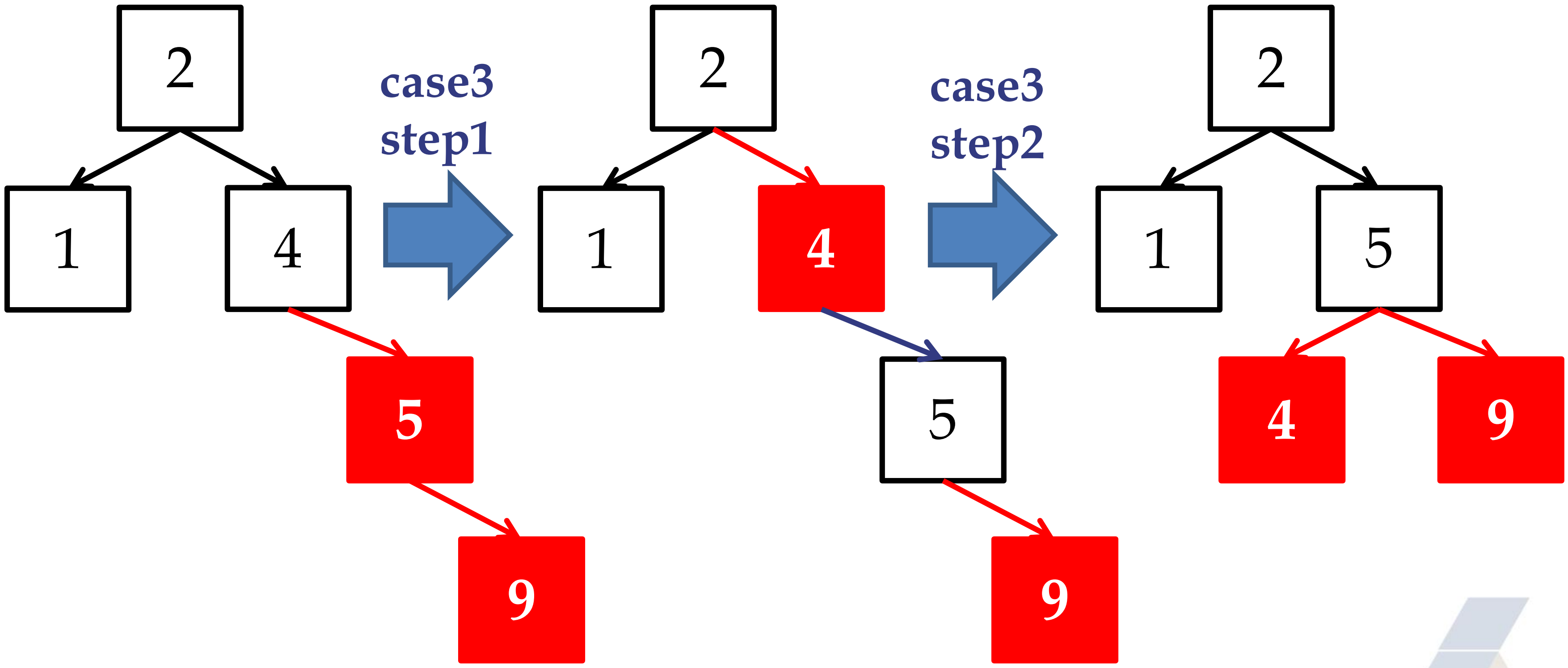
Solution 4 (Cont.)

- {2, 1, 4, **5**, 9, 3, 6, 7} → recoloring



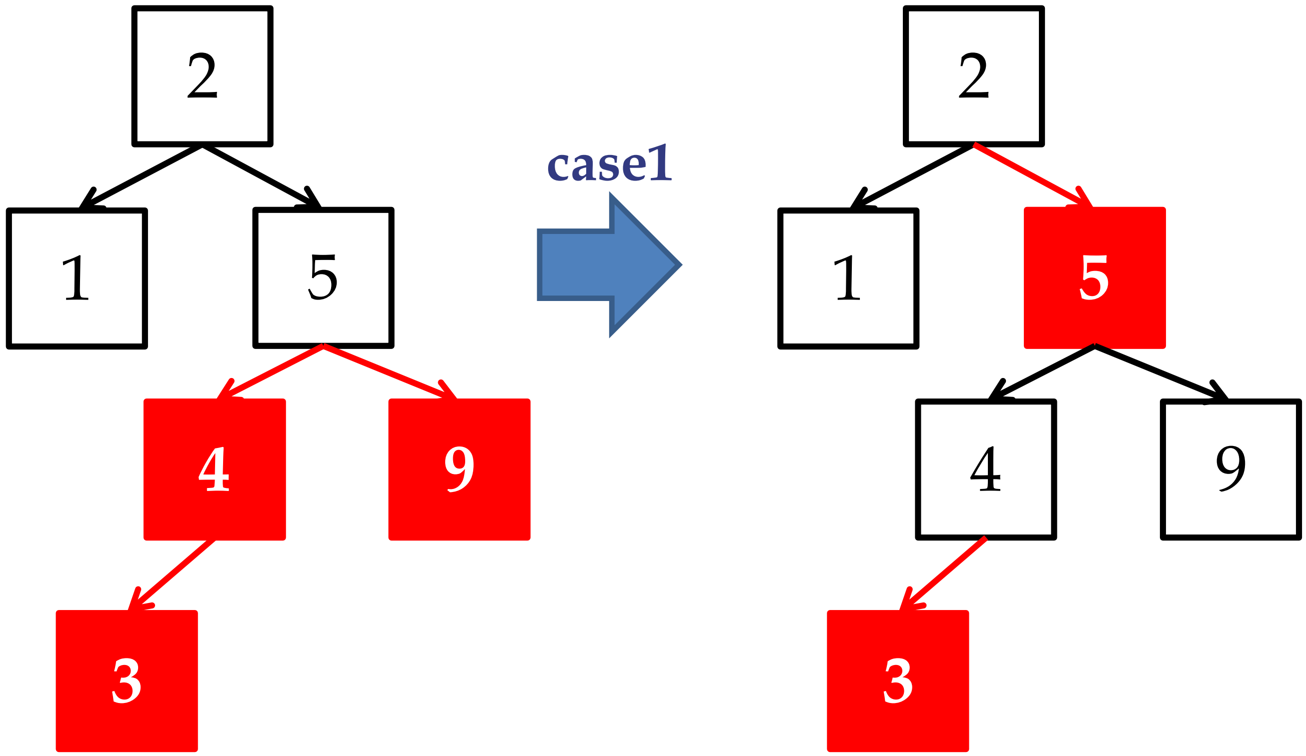
Solution 4 (Cont.)

- {2, 1, 4, 5, **9**, 3, 6, 7} → recoloring and rotation



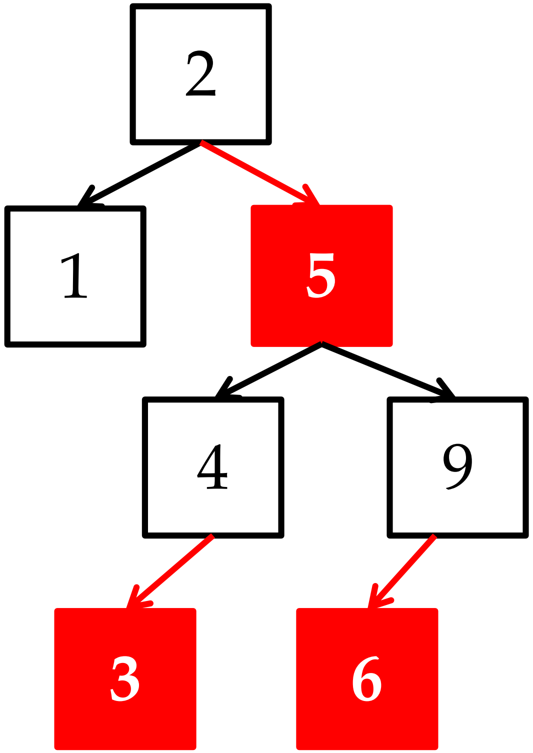
Solution 4 (Cont.)

- {2, 1, 4, 5, 9, **3**, 6, 7} → recoloring



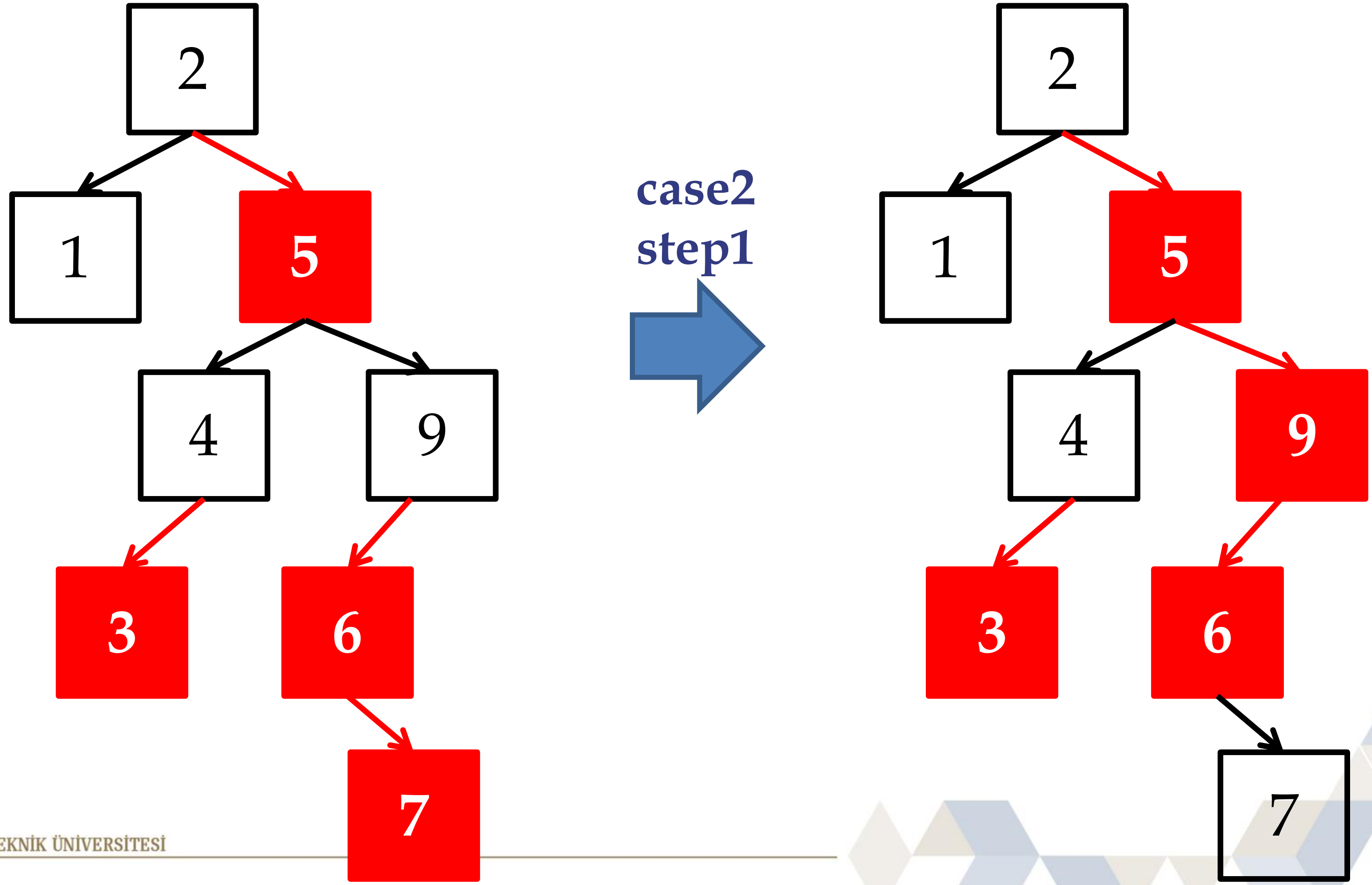
Solution 4 (Cont.)

- {2, 1, 4, 5, 9, 3, **6**, 7}



Solution 4 (Cont.)

- {2, 1, 4, 5, 9, 3, 6, 7} → recoloring



Solution 4 (Cont.)

- {2, 1, 4, 5, 9, 3, 6, 7} → 2 rotations