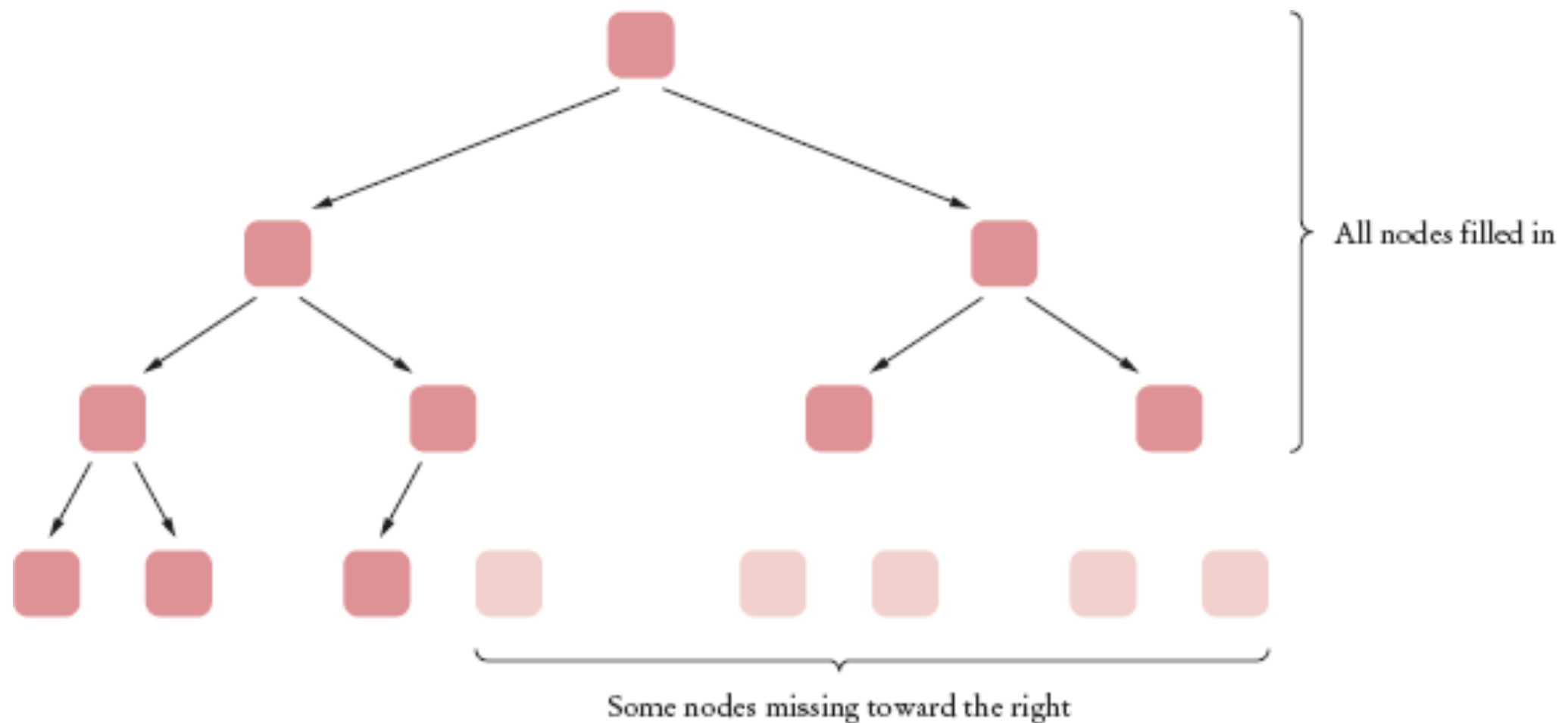


Tree-based Data Structures

Heaps:

A heap (or max-heap, in this example) is a binary tree with 2 further properties:

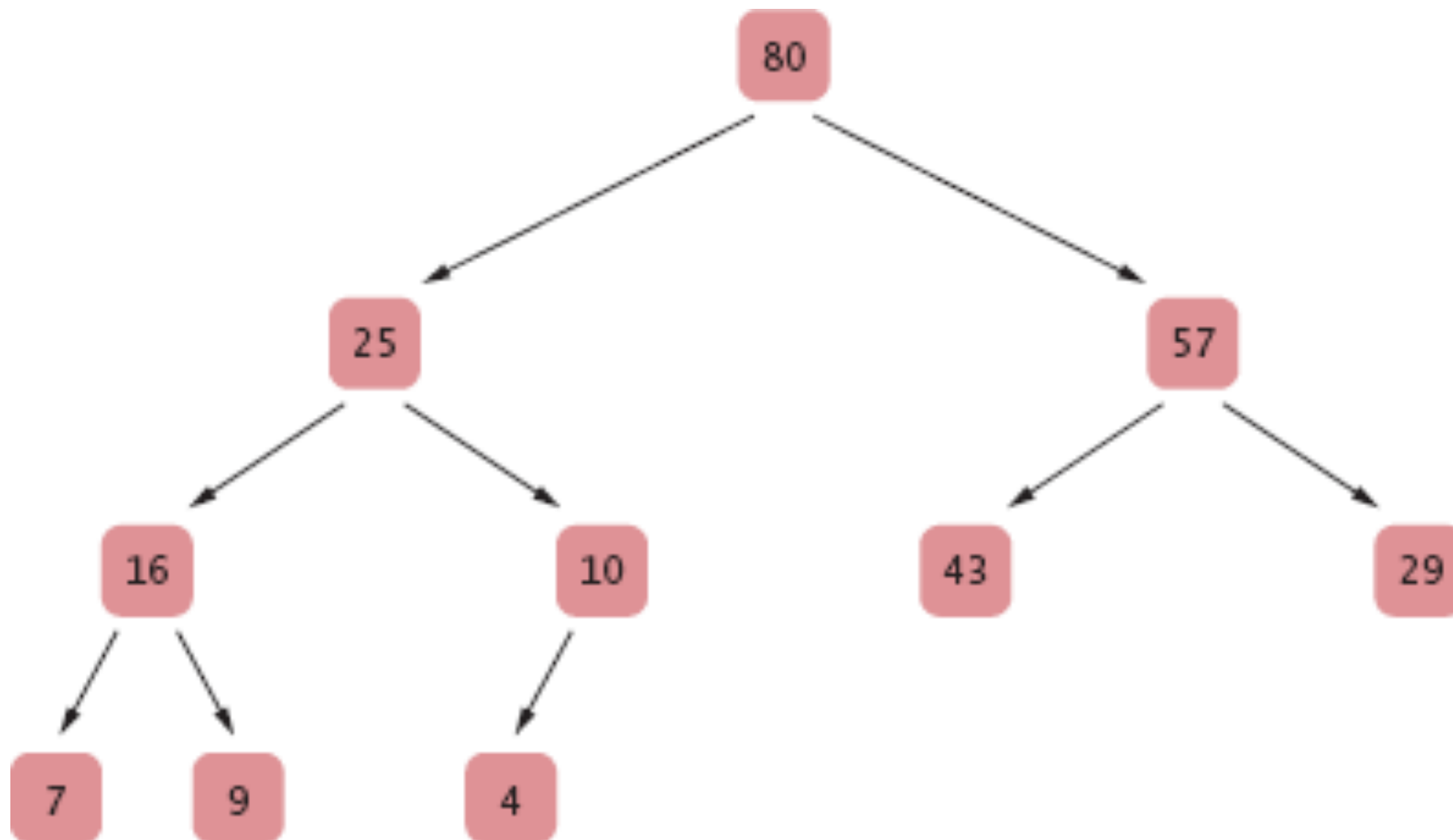
- all nodes are filled in one row (depth) at a time, from left-to-right
- each node stores a value that is at least as large as its children



Tree-based Data Structures

Heaps:

- Example heap of integers without duplicates
- Note, the largest element is at the top
- There is no notion of *sibling order*

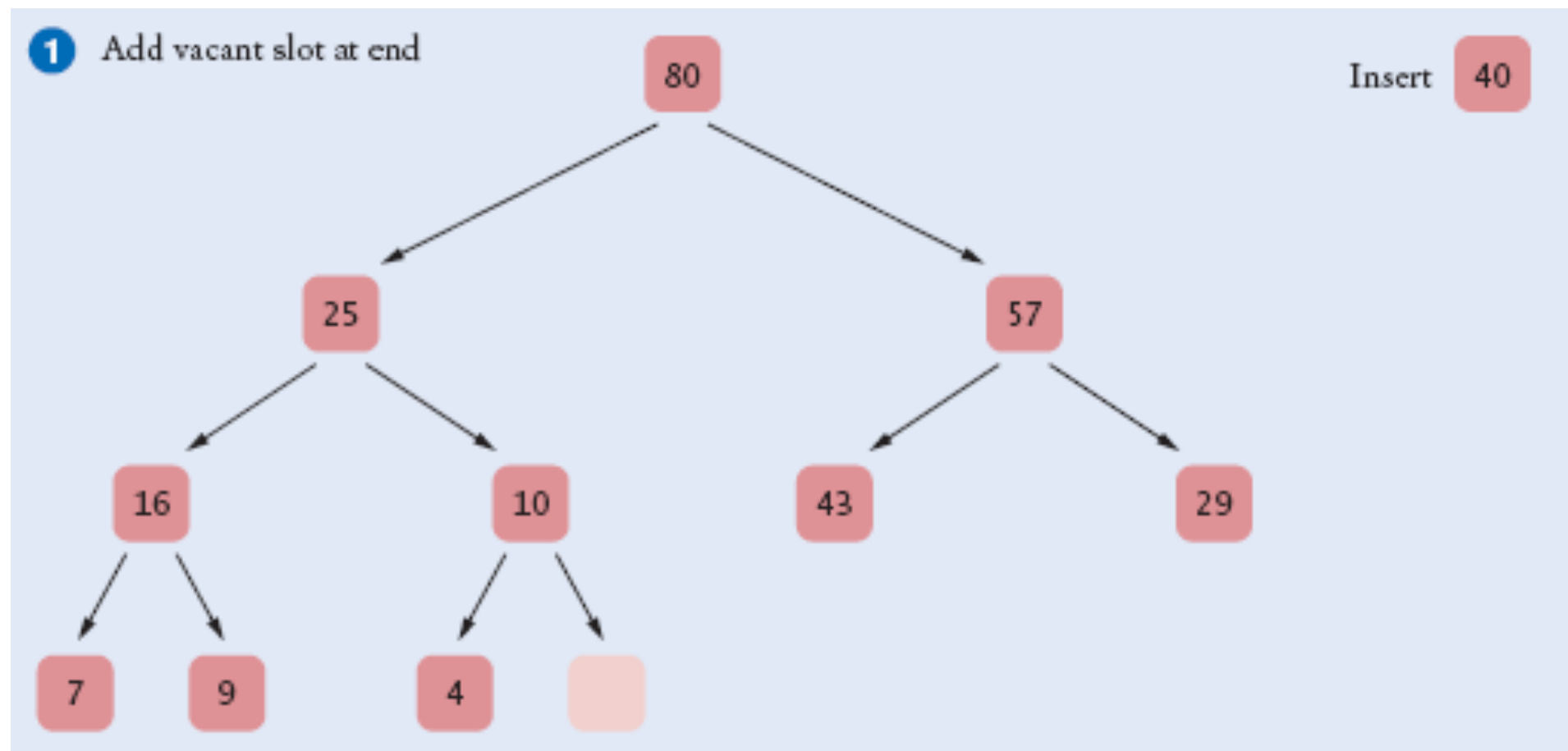


Tree-based Data Structures

Heaps:

Inserting

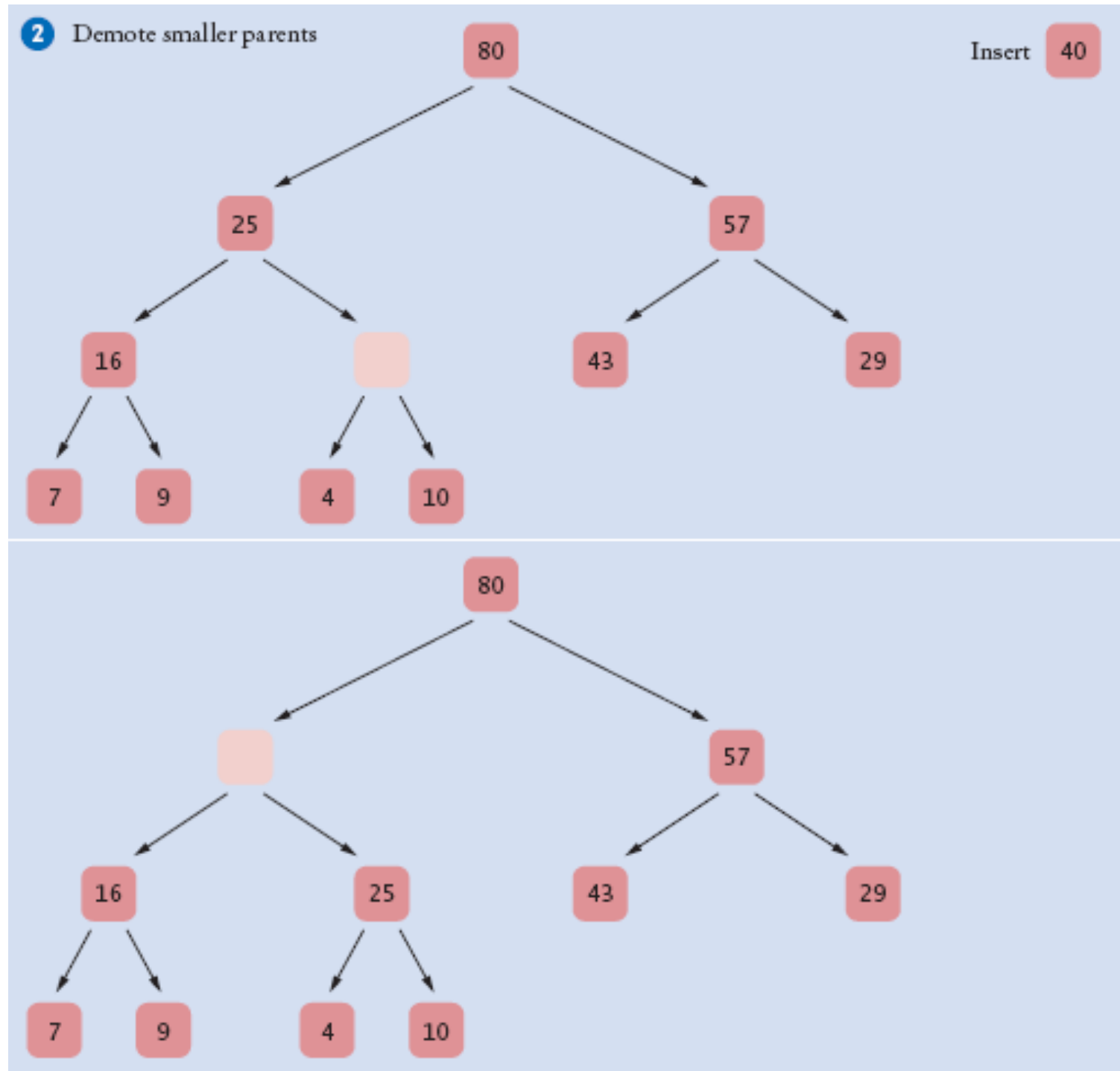
- Example: inserts 40 into the heap, above



Tree-based Data Structures

Heaps:

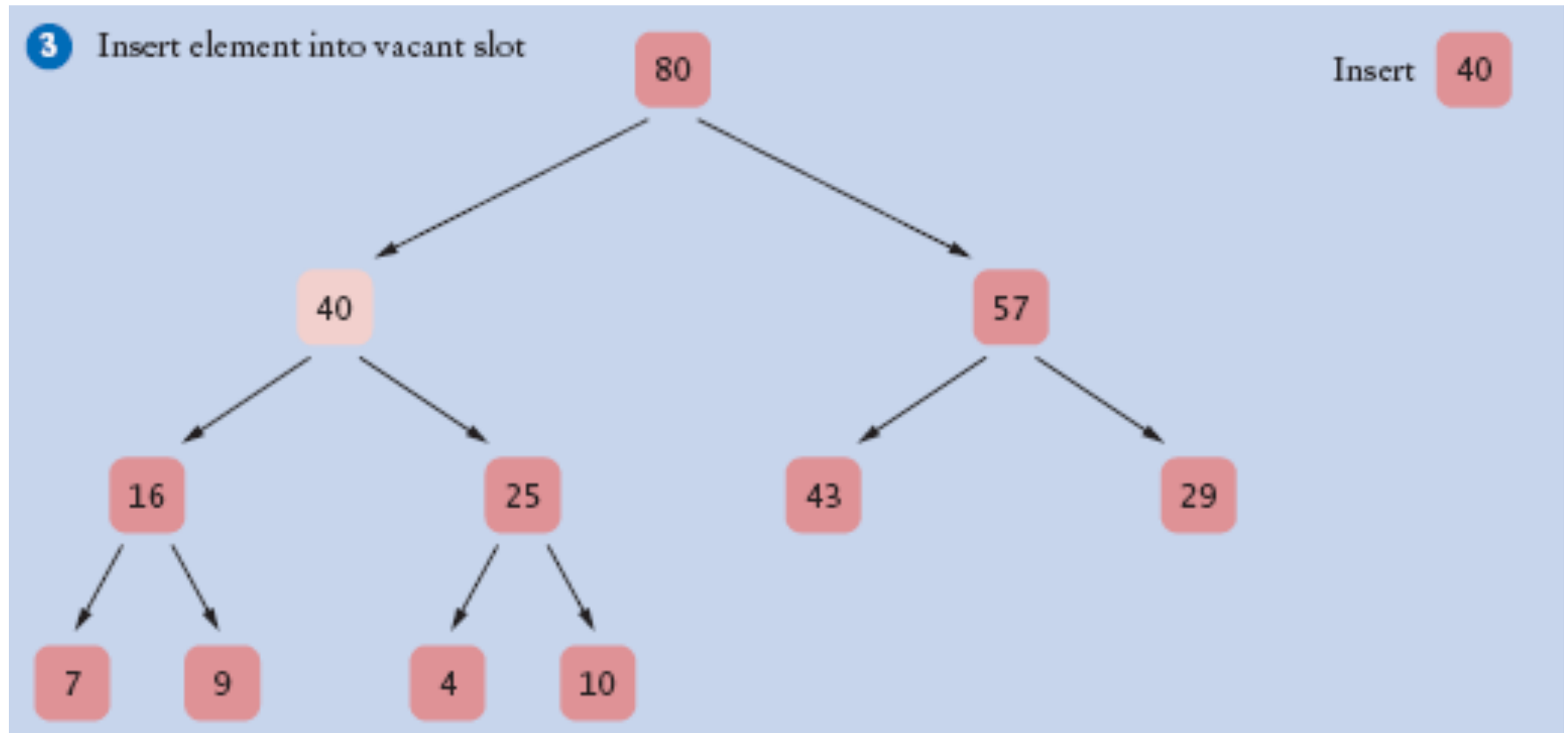
- (Inserting)



Tree-based Data Structures

Heaps:

- Inserting



Tree-based Data Structures

Heaps:

Insertion

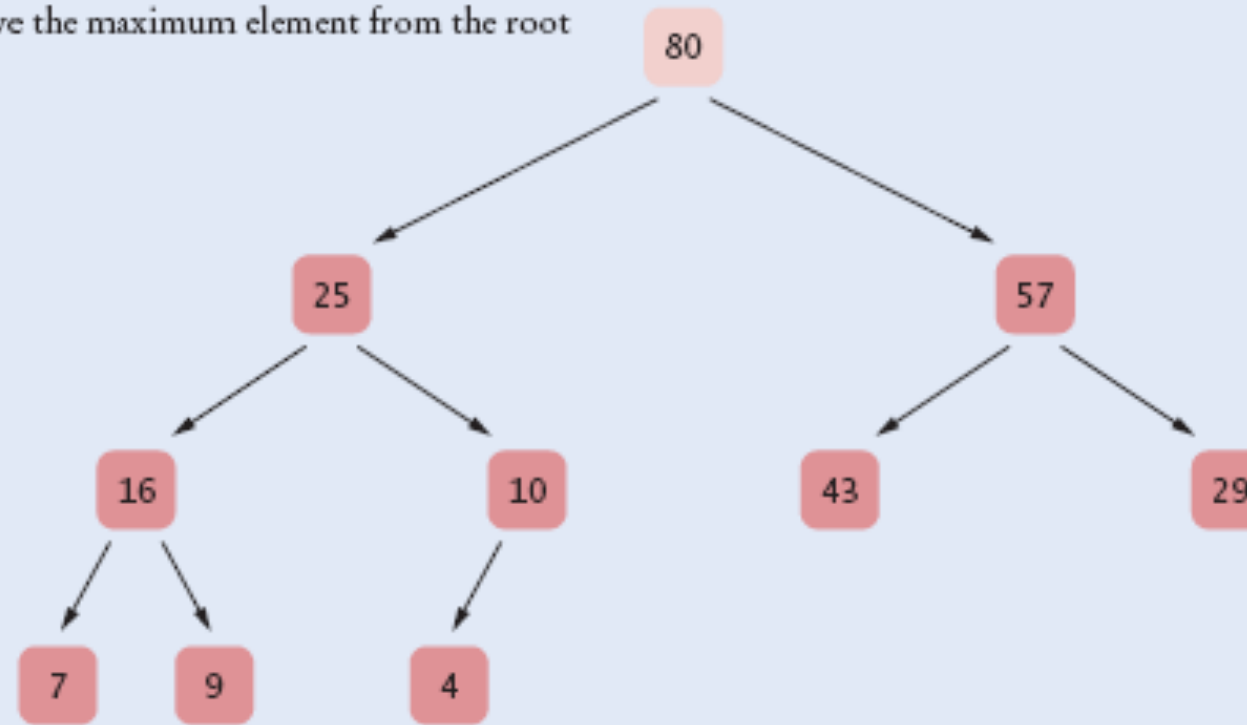
- Add a vacant slot to the end of the tree
- Demote the parent of the empty slot if it is smaller than the element
- Repeat the demotion until the parent is larger than the inserting element
- Special case: empty heap

Tree-based Data Structures

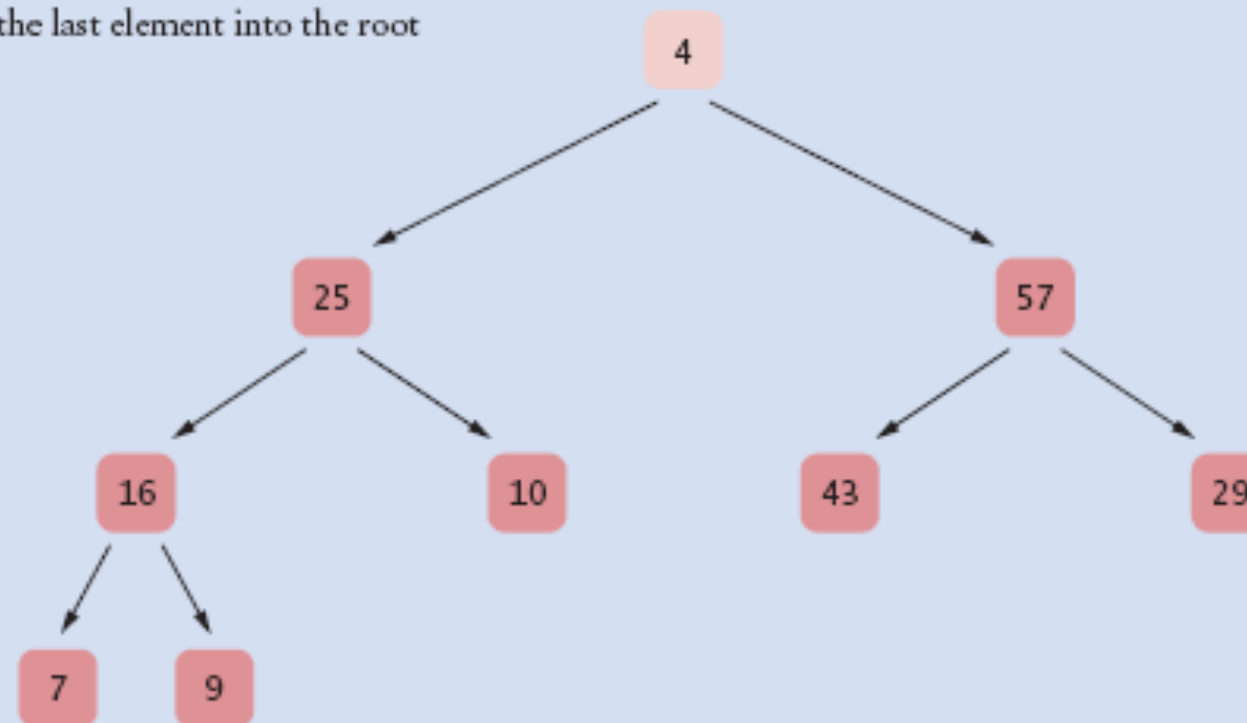
Heaps:

- Removal the root

1 Remove the maximum element from the root

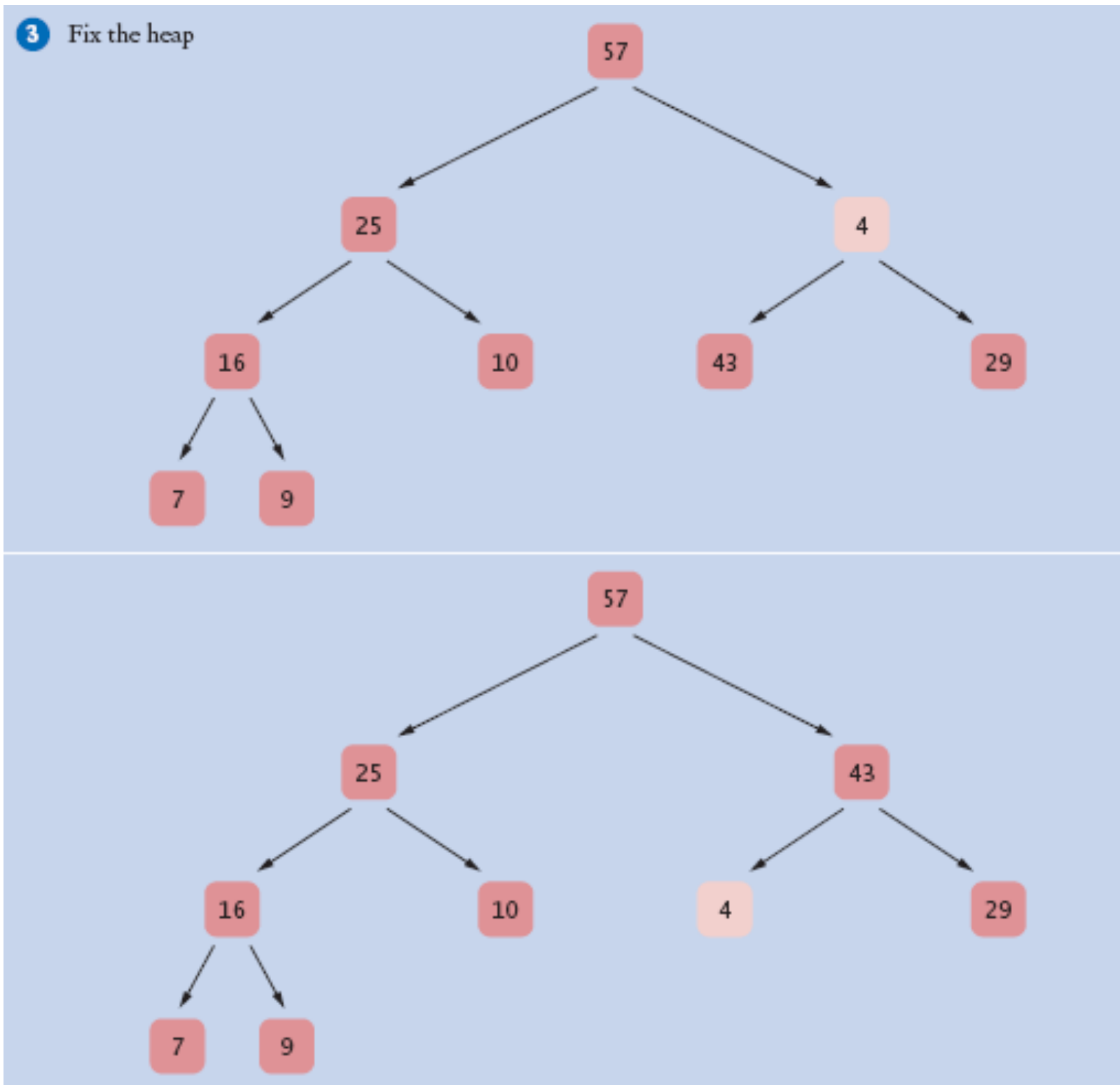


2 Move the last element into the root



Tree-based Data Structures

Heaps:



Tree-based Data Structures

Heaps:

Remove the root

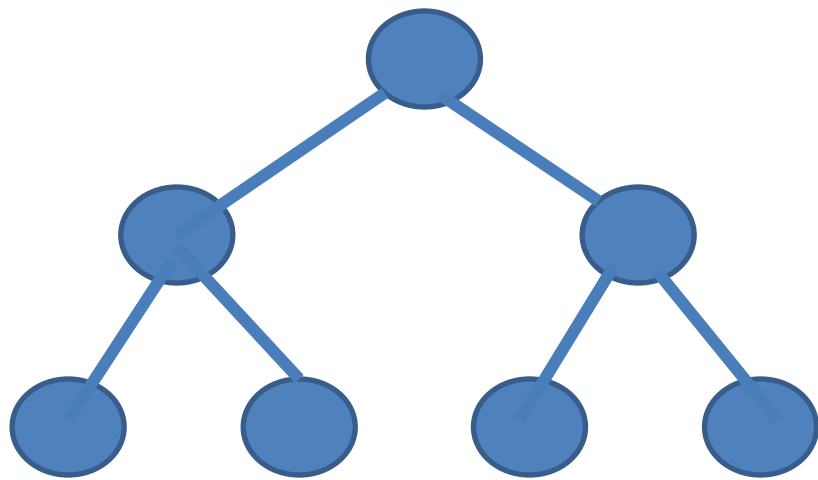
- Replace the root with the "last" leaf element
- Fix the heap property by downheaping the element at the root

Tree-based Data Structures

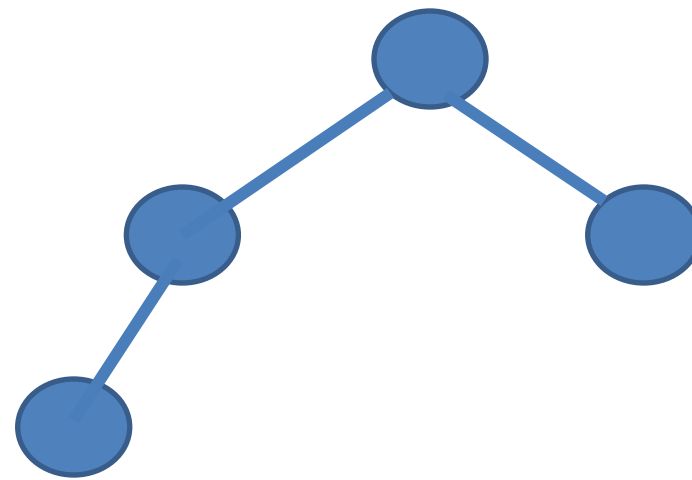
Heaps:

Cost

- Insertion or removal requires h swaps at most
- Tree of height h contains at least $2^{(h-1)}$ nodes, but less than 2^h nodes
- $2^{(h-1)} \leq n < 2^h \rightarrow h-1 \leq \log_2(n) < h \rightarrow \log_2(n) < h \leq \log_2(n)+1$
- Insertion and removal is $O(\log(n))$ operations



Height = h
Nodes $n = 2^h - 1$ (largest)

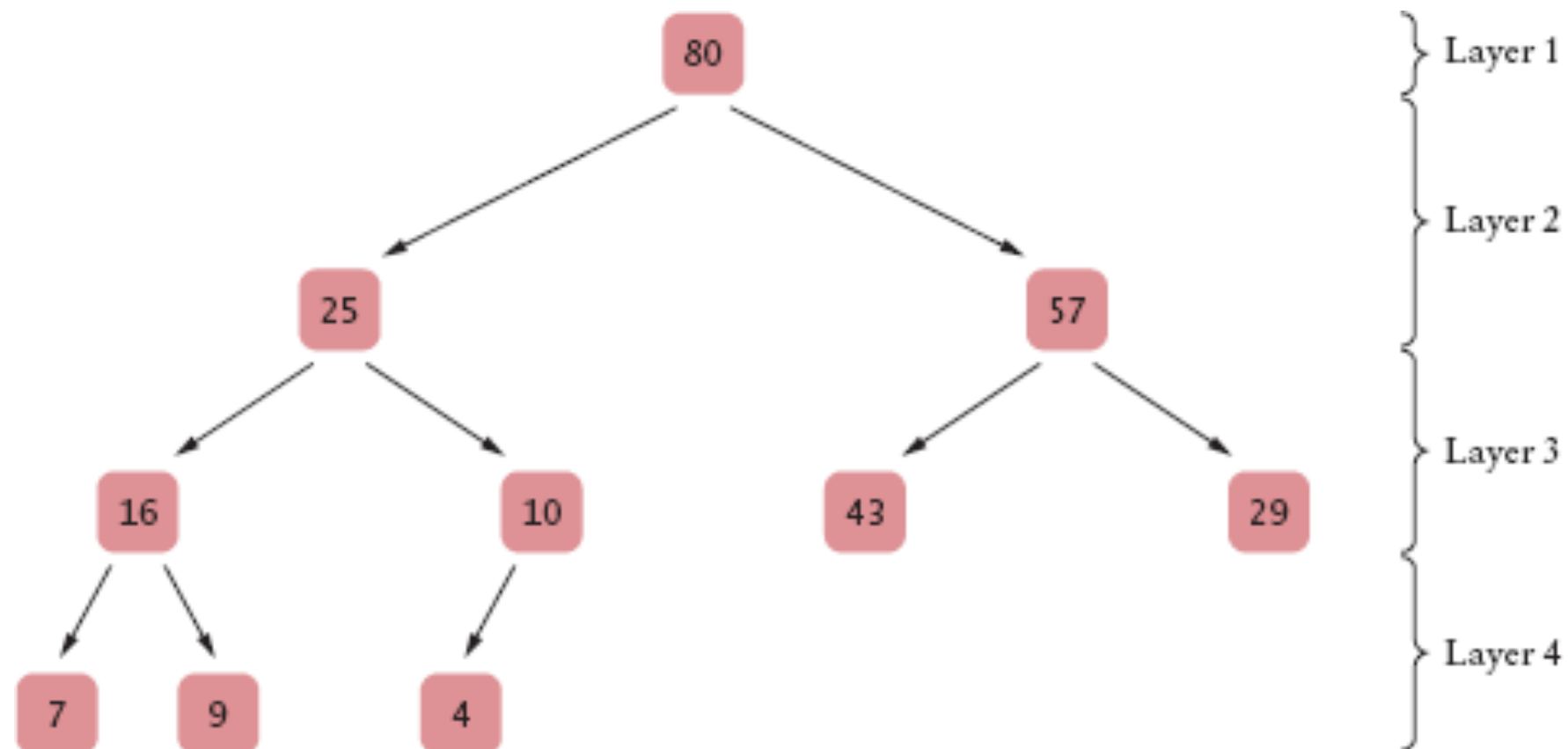


Height = h
Nodes $n = 2^{(h-1)}$ (smallest)

Tree-based Data Structures

Heaps: Representation

- Because of the heap's shape, node values can be stored in an array



- Elements are stored from top to bottom, from left to right
- Example: above heap is stored as:

80	25	57	16	10	43	29	7	9	4
----	----	----	----	----	----	----	---	---	---