Idea, searching and inserting in B+-tree

Overview over this video

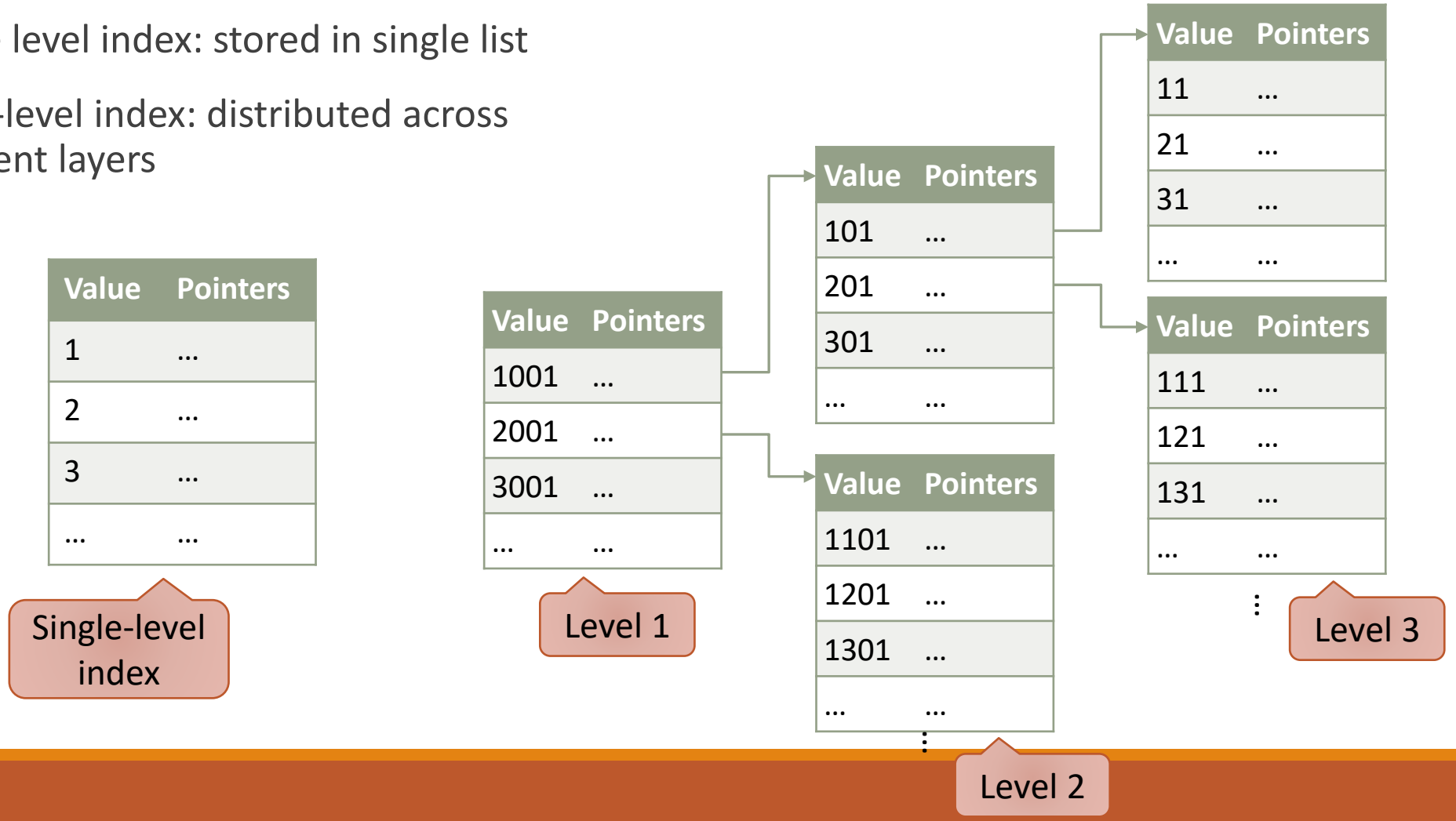
The video will cover the basics of B+-trees as well as

- Searching for values
- Inserting pairs of values and pointers

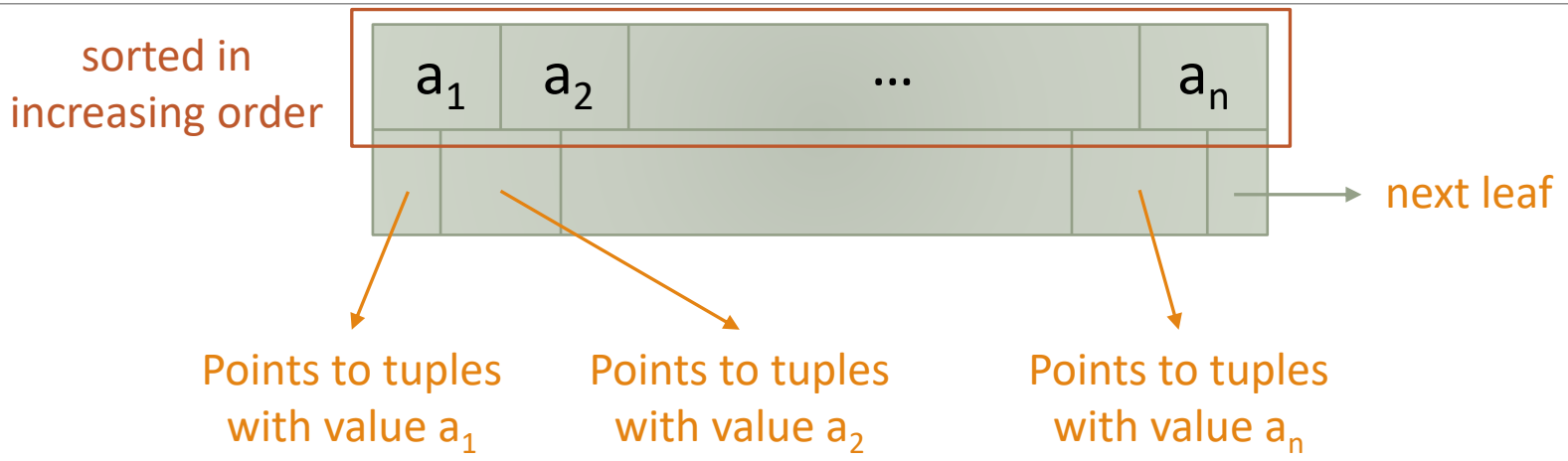
Single Level vs Multi-Level Indexes

Single level index: stored in single list

Multi-level index: distributed across different layers



B+ Tree: Leaves (Idea)



n = chosen such that node fits into a single disk block

Example:

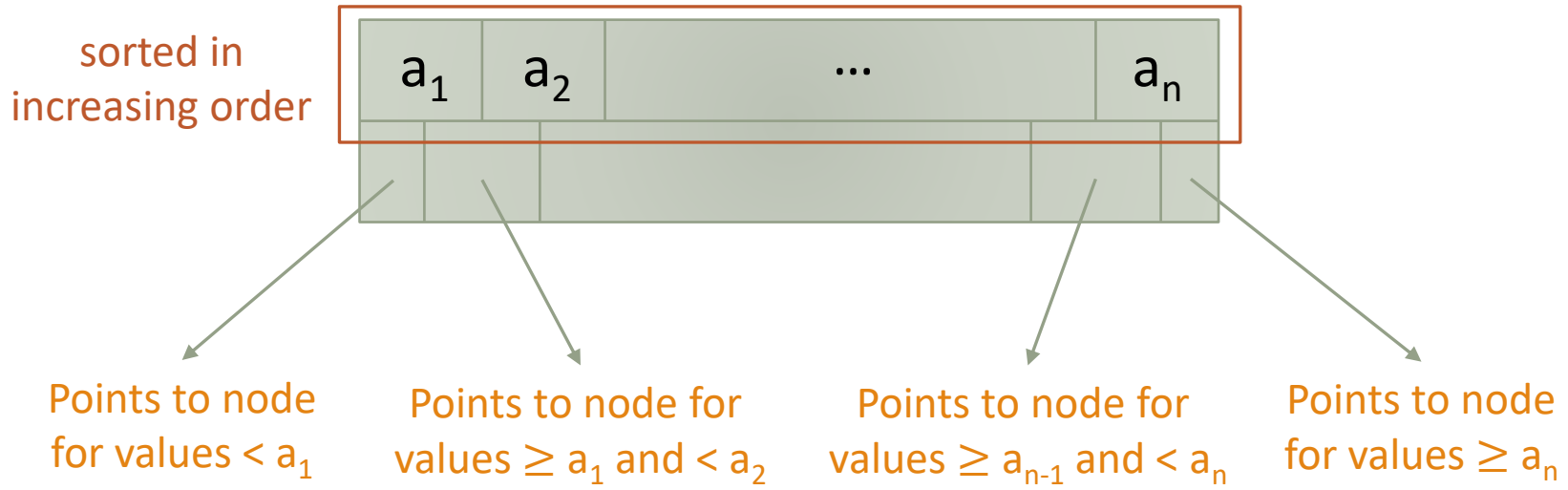
- Disk block size = 512 byte
- Values: 4 byte integers
- Pointers: 8 bytes

n = 42

$(512 \geq 42\ (8+4)+8$

$512 < 43\ (8+4)+8)$

B+ Tree: Inner Nodes (Idea)



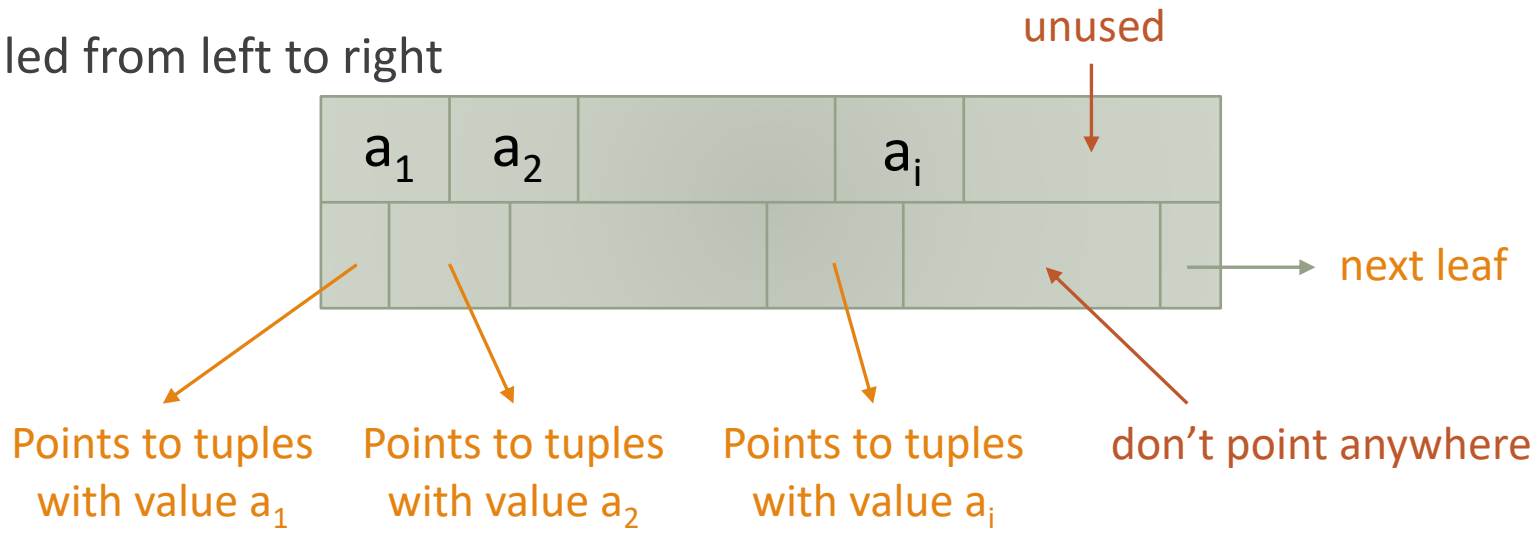
Pointers point to B+ tree nodes at level below

n = chosen as before

B+ Tree: Leaves (Actually)

Not all of the fields have to be used

Fields are filled from left to right

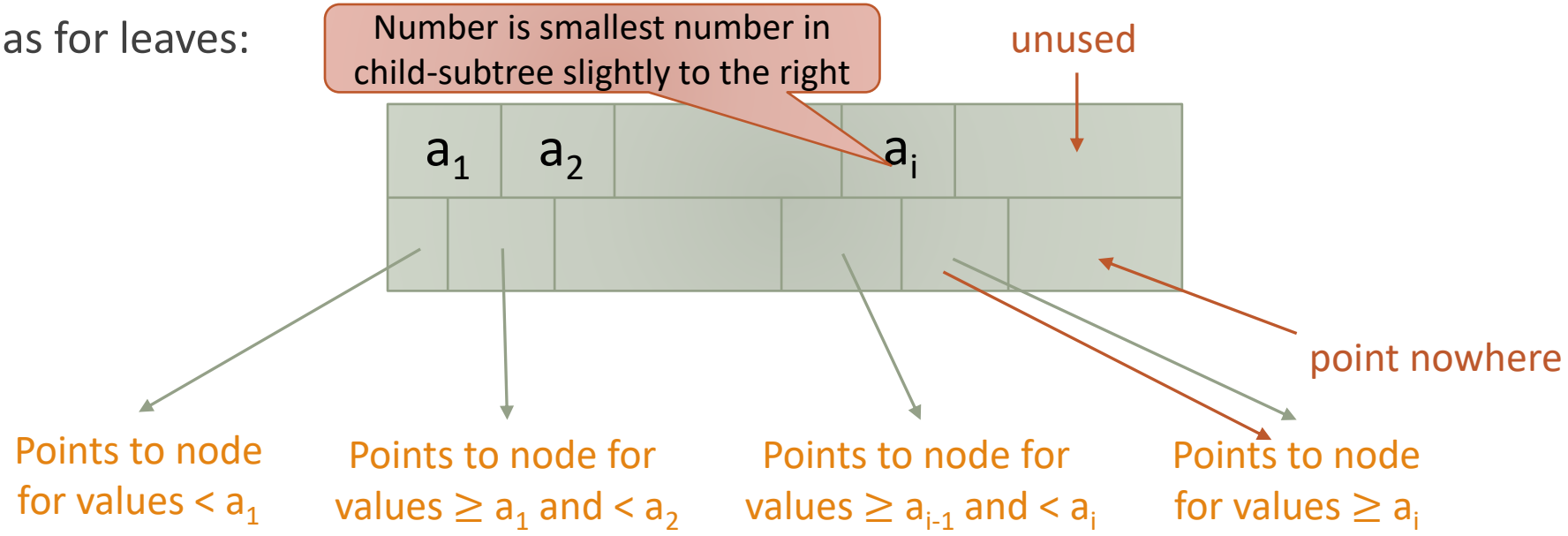


Ensure that at least $\left\lfloor \frac{n+1}{2} \right\rfloor$ pointers are used (unless this is the only leaf)

To follow the online tool for this (will show it soon), we count the next leaf pointer as a pointer (fairly sensible), even if there are none (less sensible)

B+ Tree: Inner Nodes (Actually)

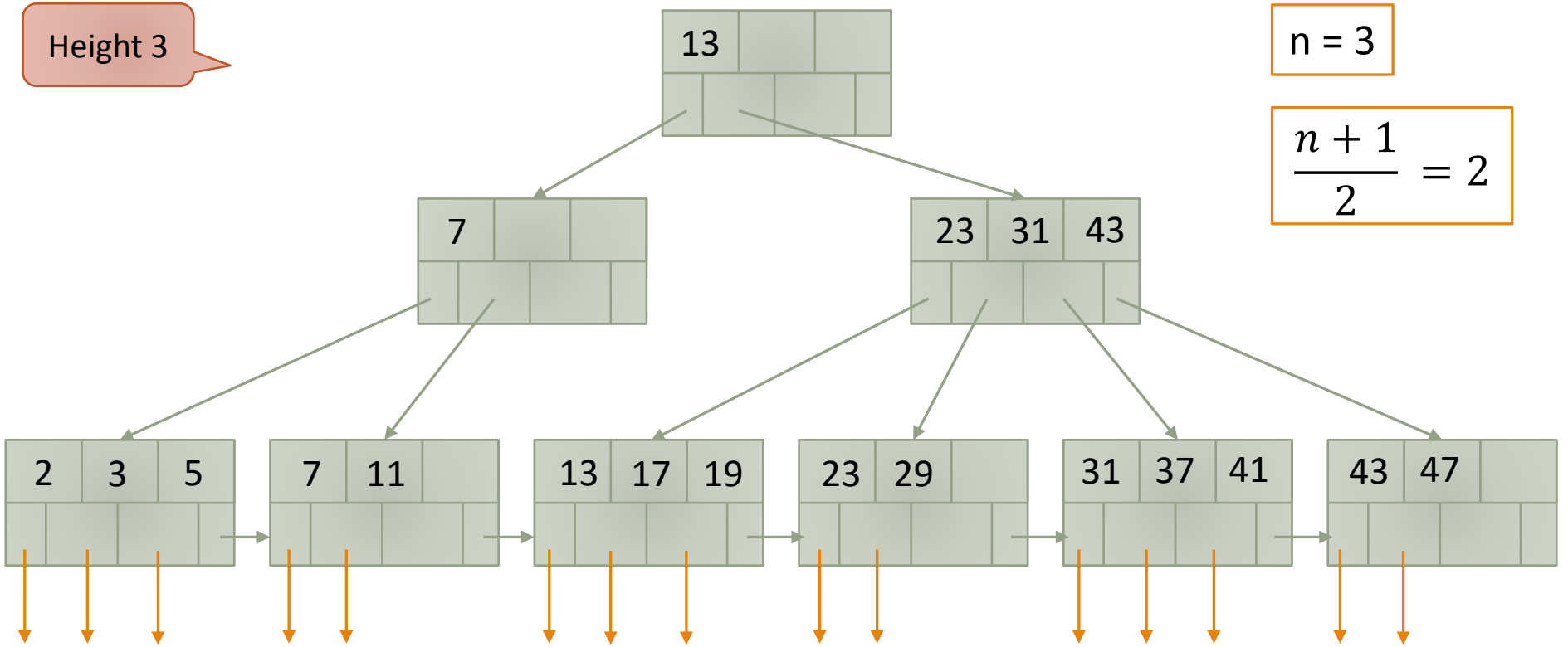
Similar as for leaves:



Ensure that at least $\left\lceil \frac{n+1}{2} \right\rceil$ pointers are used

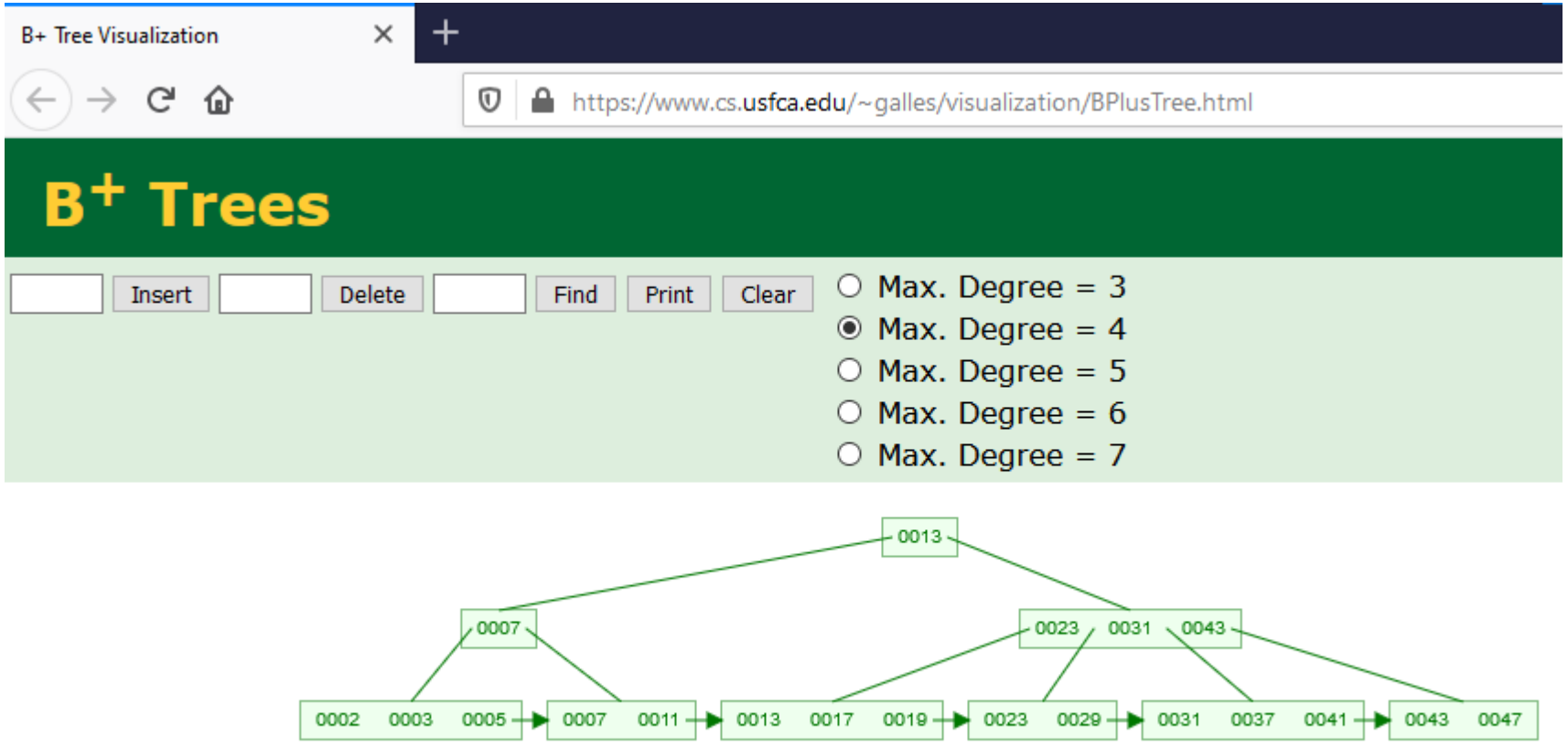
Exception: root must use ≥ 2 pointers

A B+ Tree Index



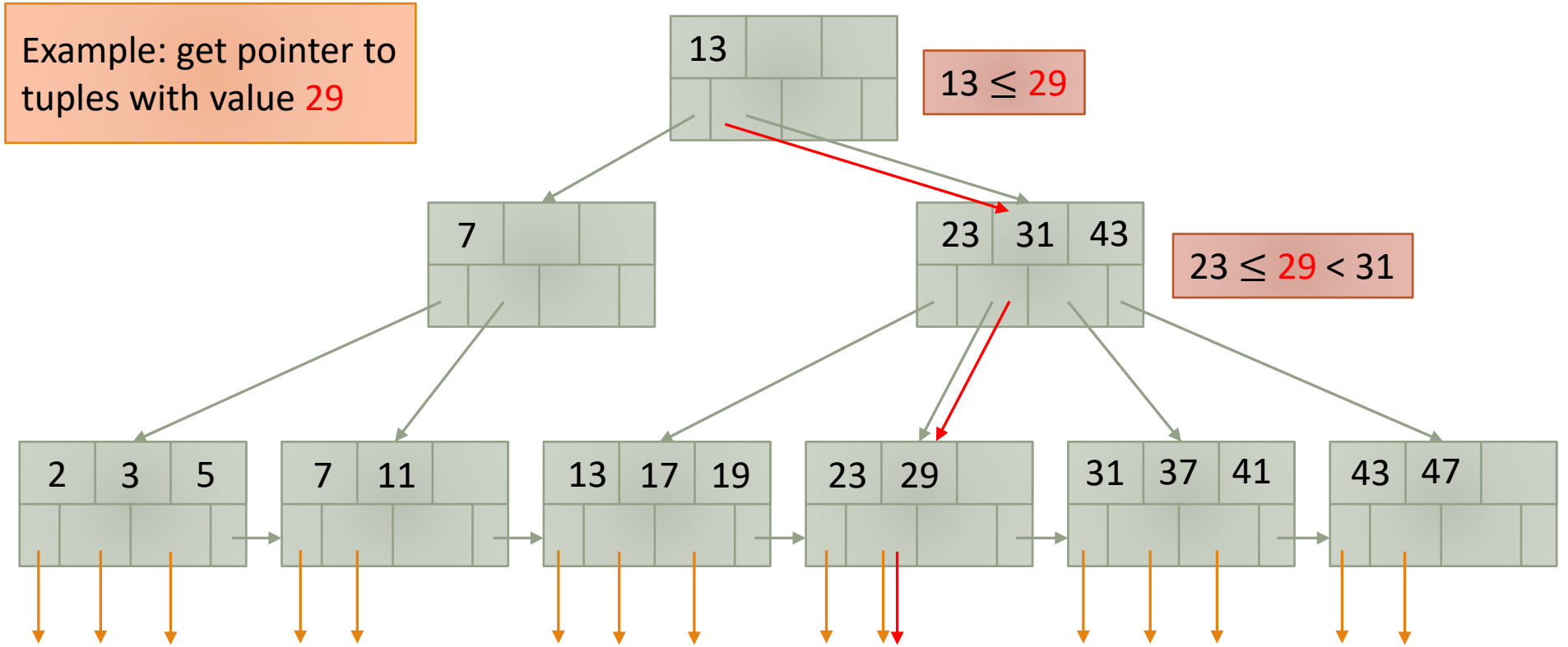
(Example from ”Database Systems: The Complete Book”)

Online version



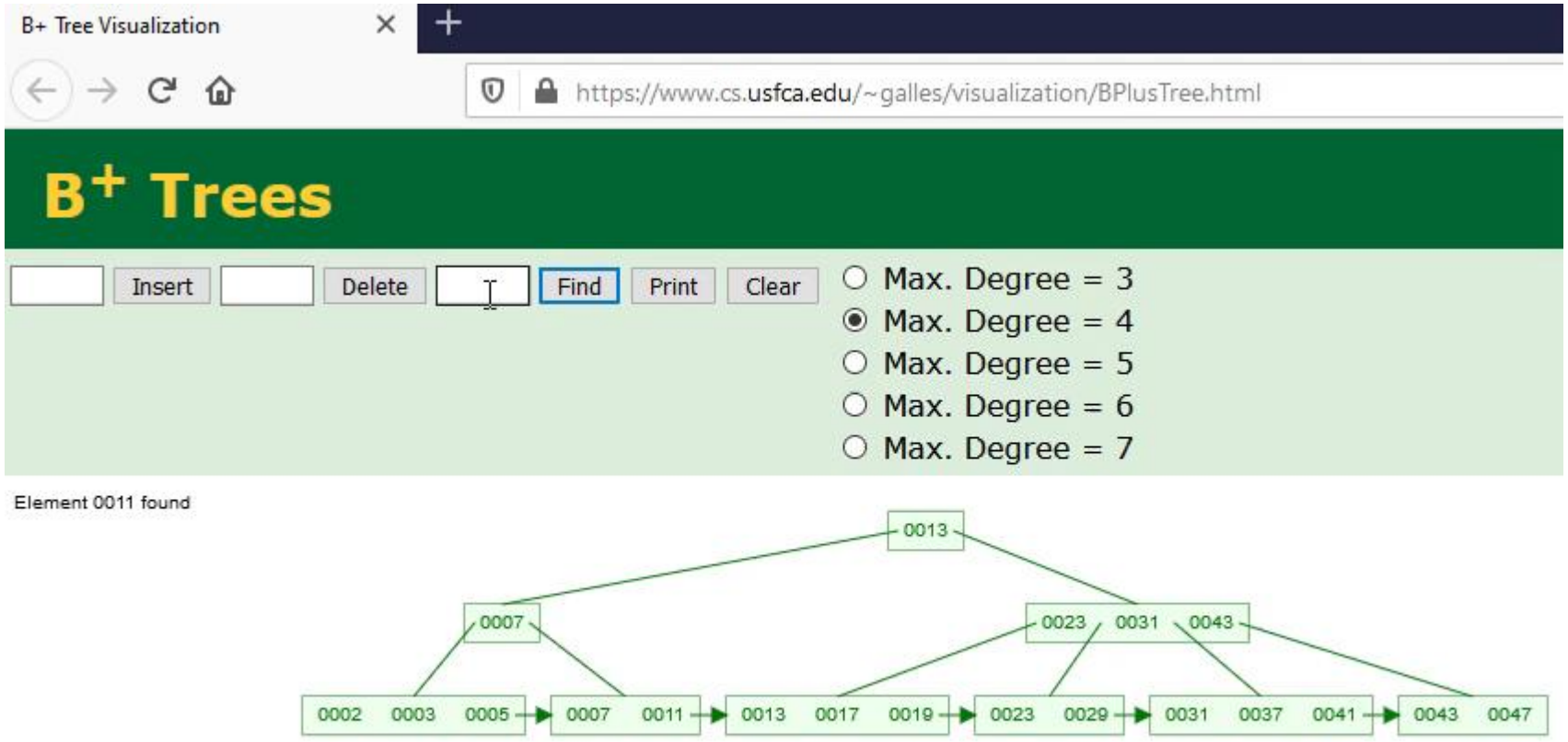
Looking Up Values

Example: get pointer to tuples with value 29

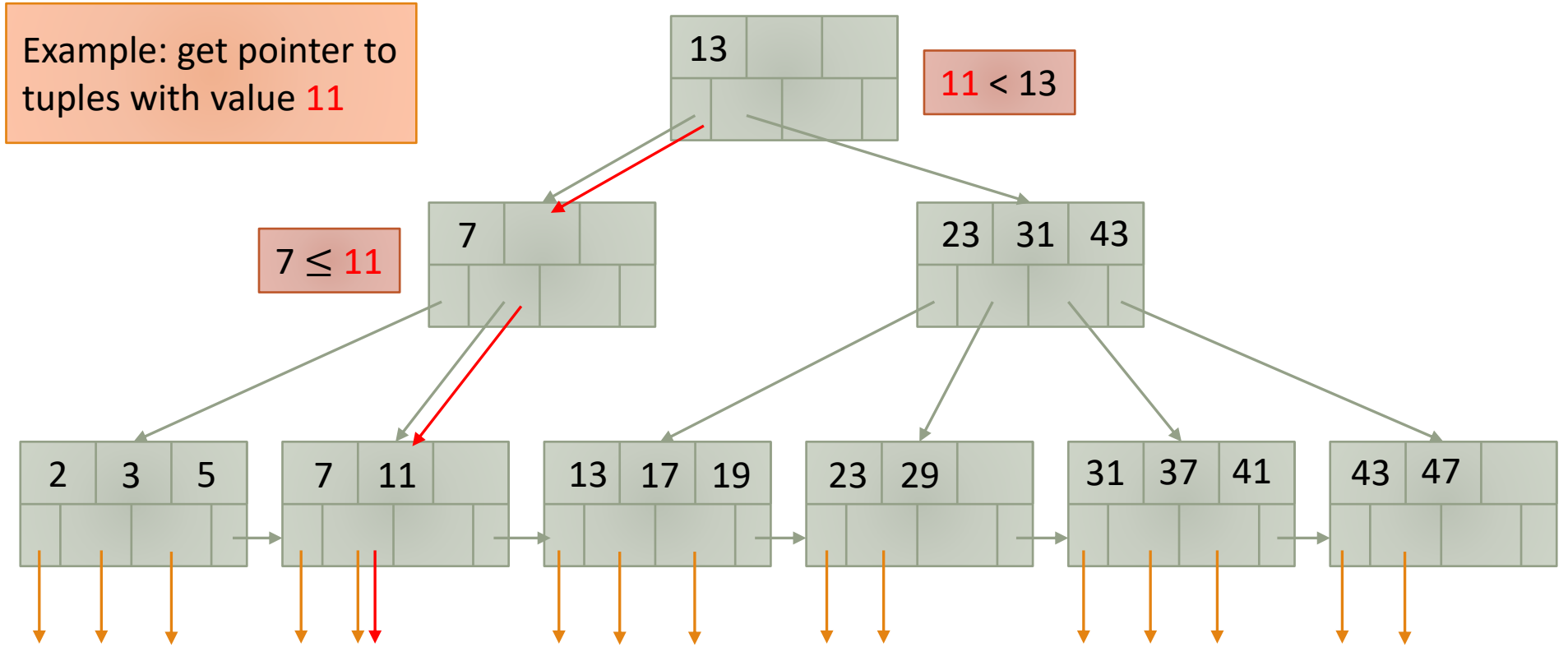


(Example from ”Database Systems: The Complete Book”)

Online version of looking for 29

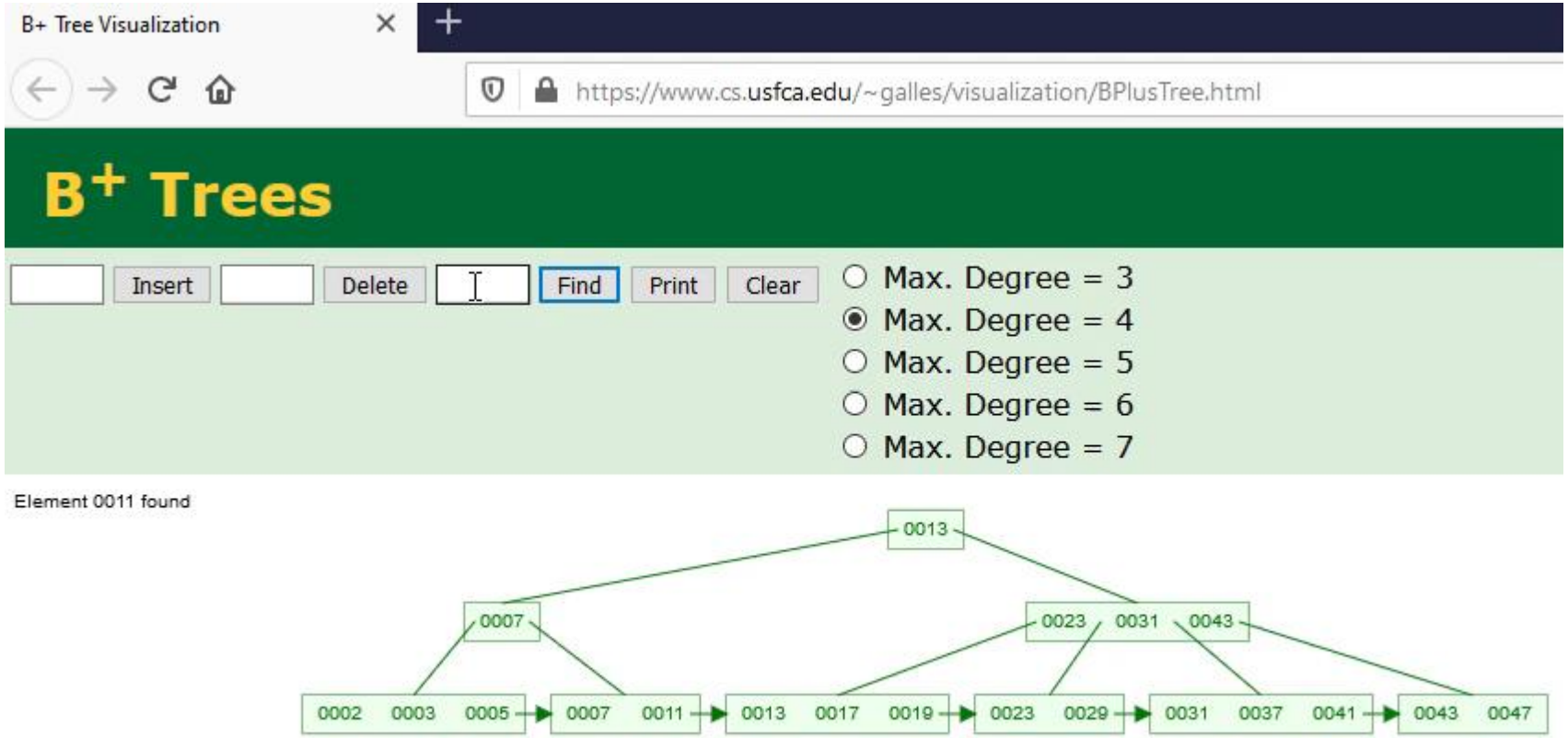


Example 2

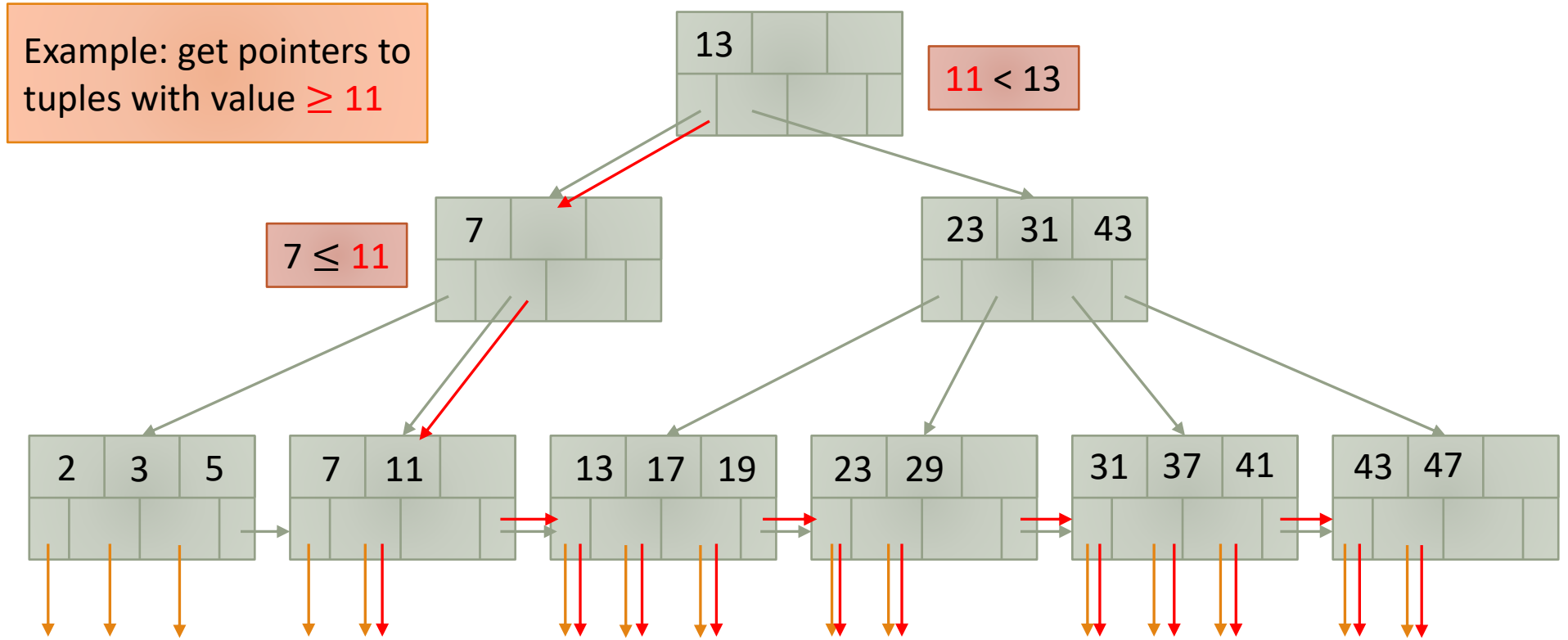


(Example from ”Database Systems: The Complete Book”)

Online version of looking for 11



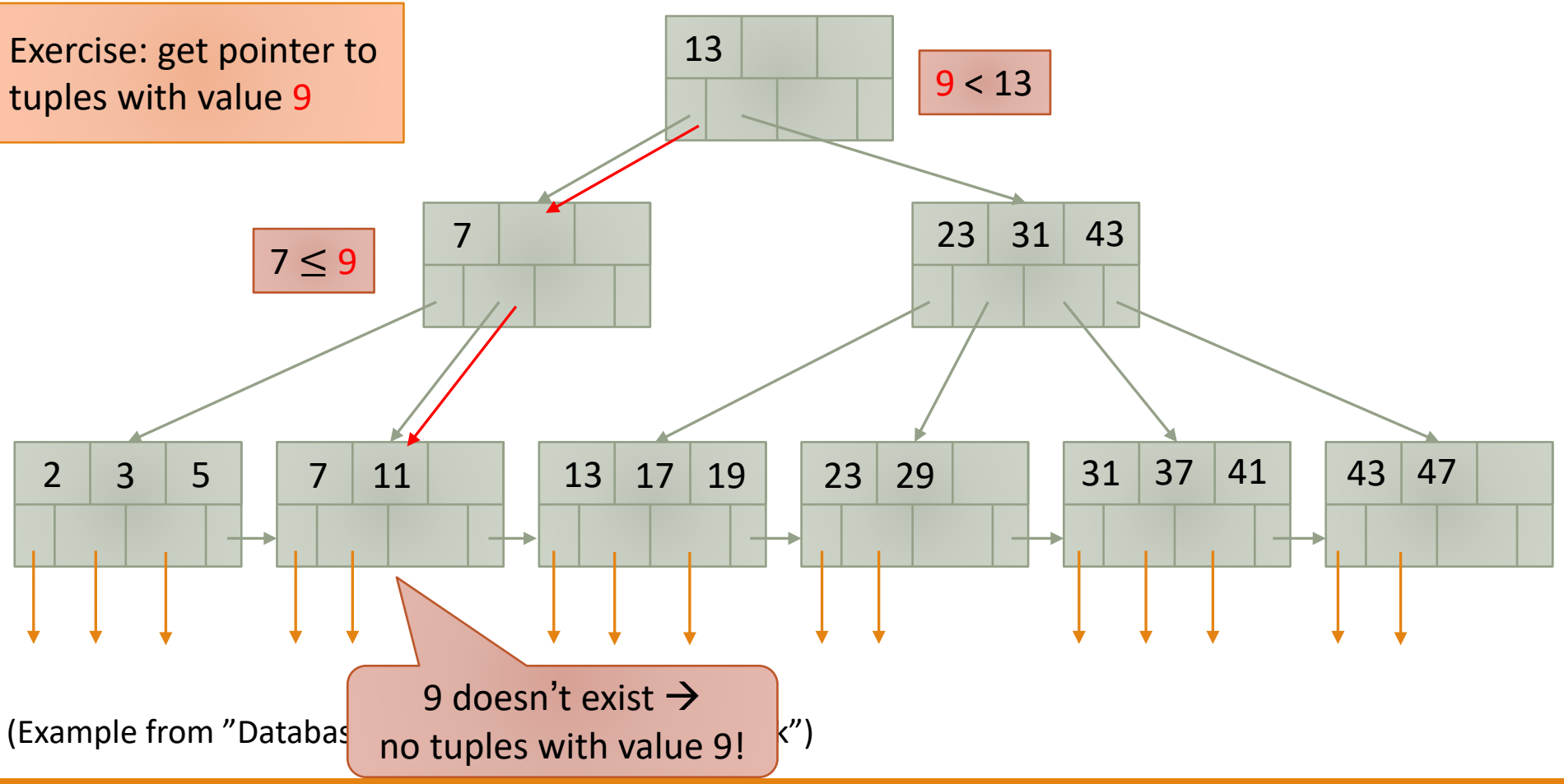
Example 3



(Example from ”Database Systems: The Complete Book”)

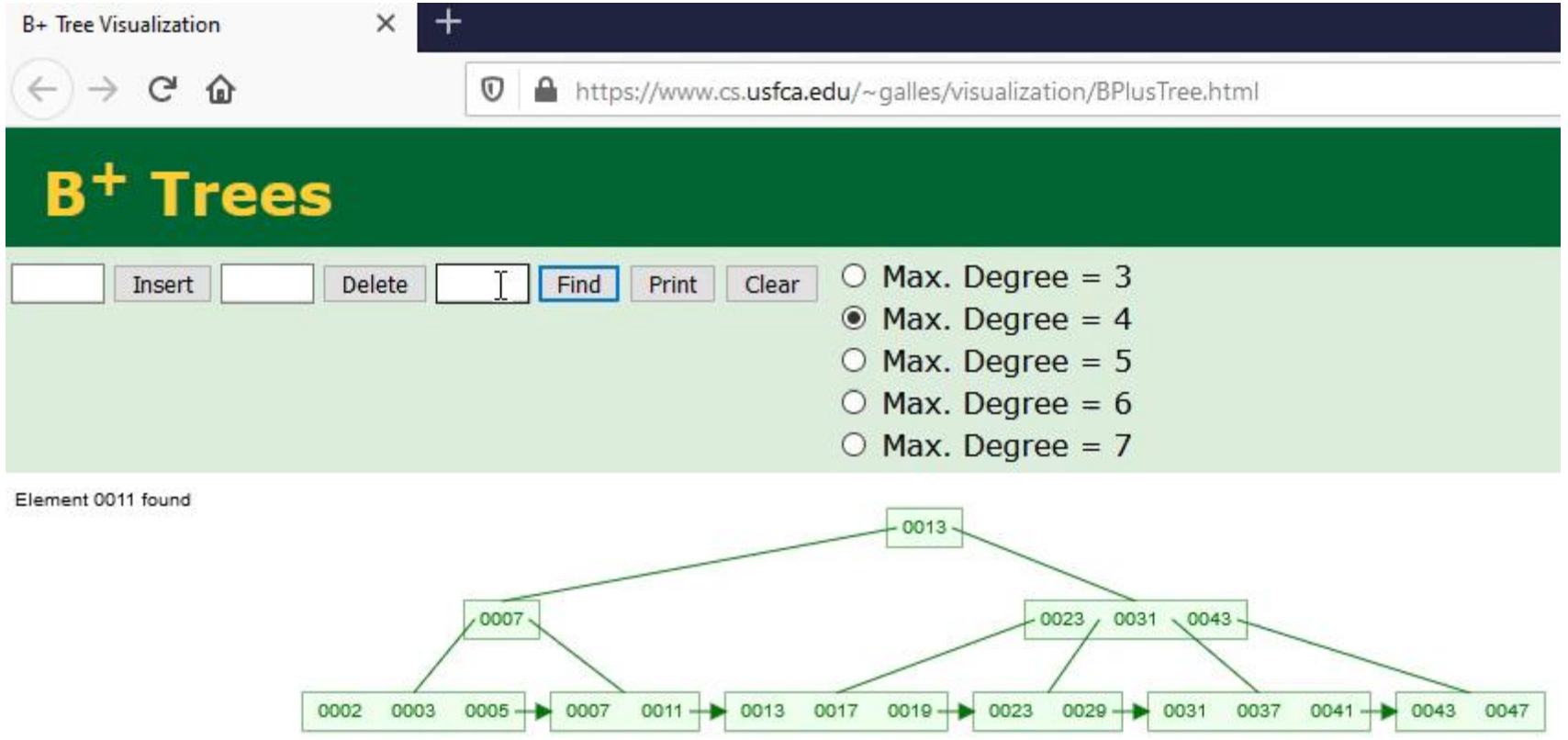
Example 4

Exercise: get pointer to tuples with value 9



(Example from "Databas k")

Online version of looking for 9

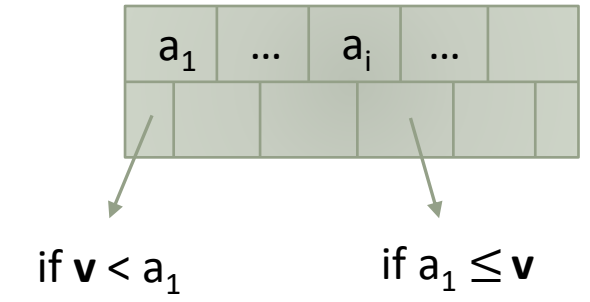


Looking Up A Value: Summary

Goal: find the pointer to the rows with value **v**

Procedure:

- Start at the root of the B+ tree
- While the current node is a non-leaf node:
  - If **v** < a_1, proceed to the first child of the node
  - Otherwise find the **largest i** with $a_i \leq$ **v** and proceed to the associated child node
- If the current node is a leaf:
  - If **v** occurs in the leaf, follow the associated pointer
  - If **v** does not occur in the leaf, return "**v** does not exist in index"

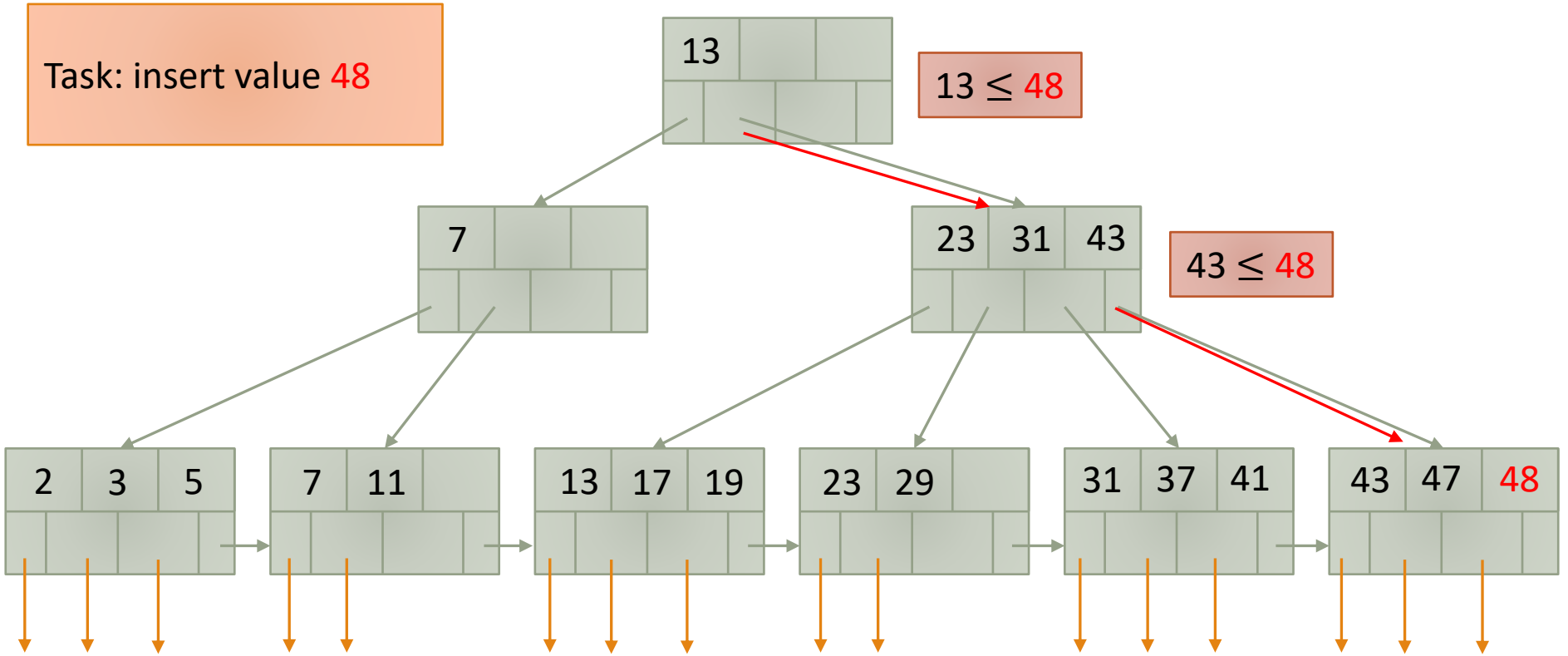


Time for a disk operation

Running time: $O(h \times \log_2 n)$ "real" running time $O(h \times D)$

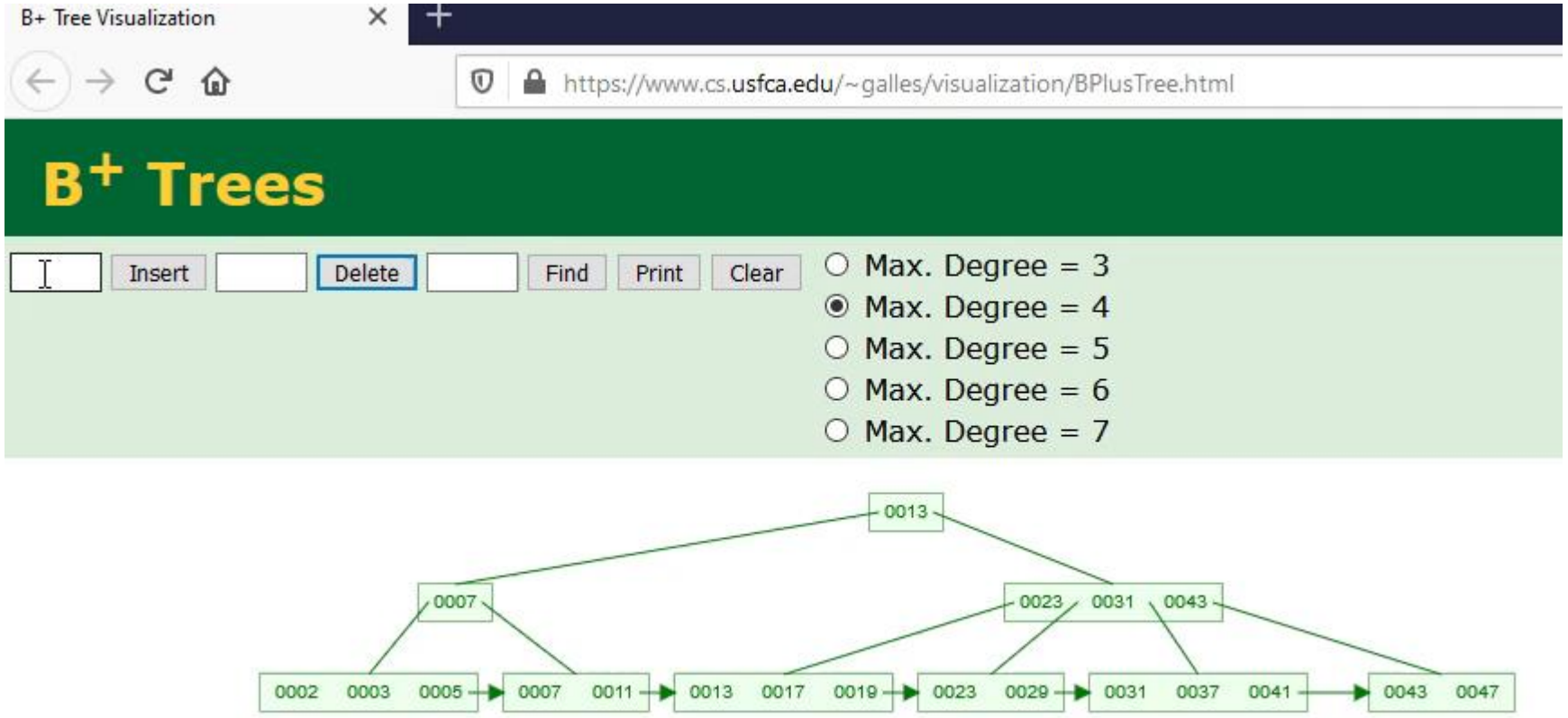
Height of the B+ tree

Insertions of Value/Pointer Pairs

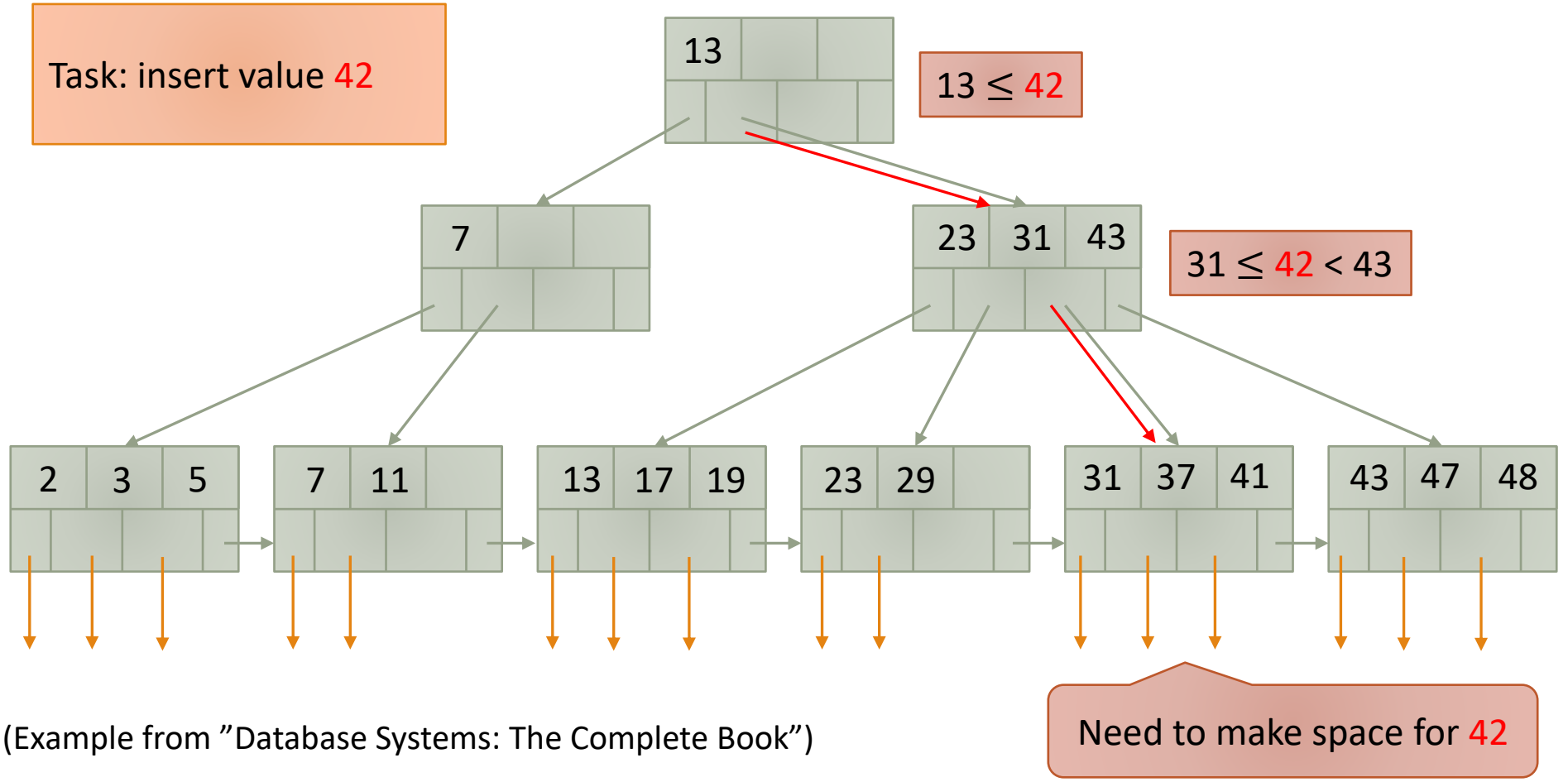


(Example from ”Database Systems: The Complete Book”)

Online version of inserting 48

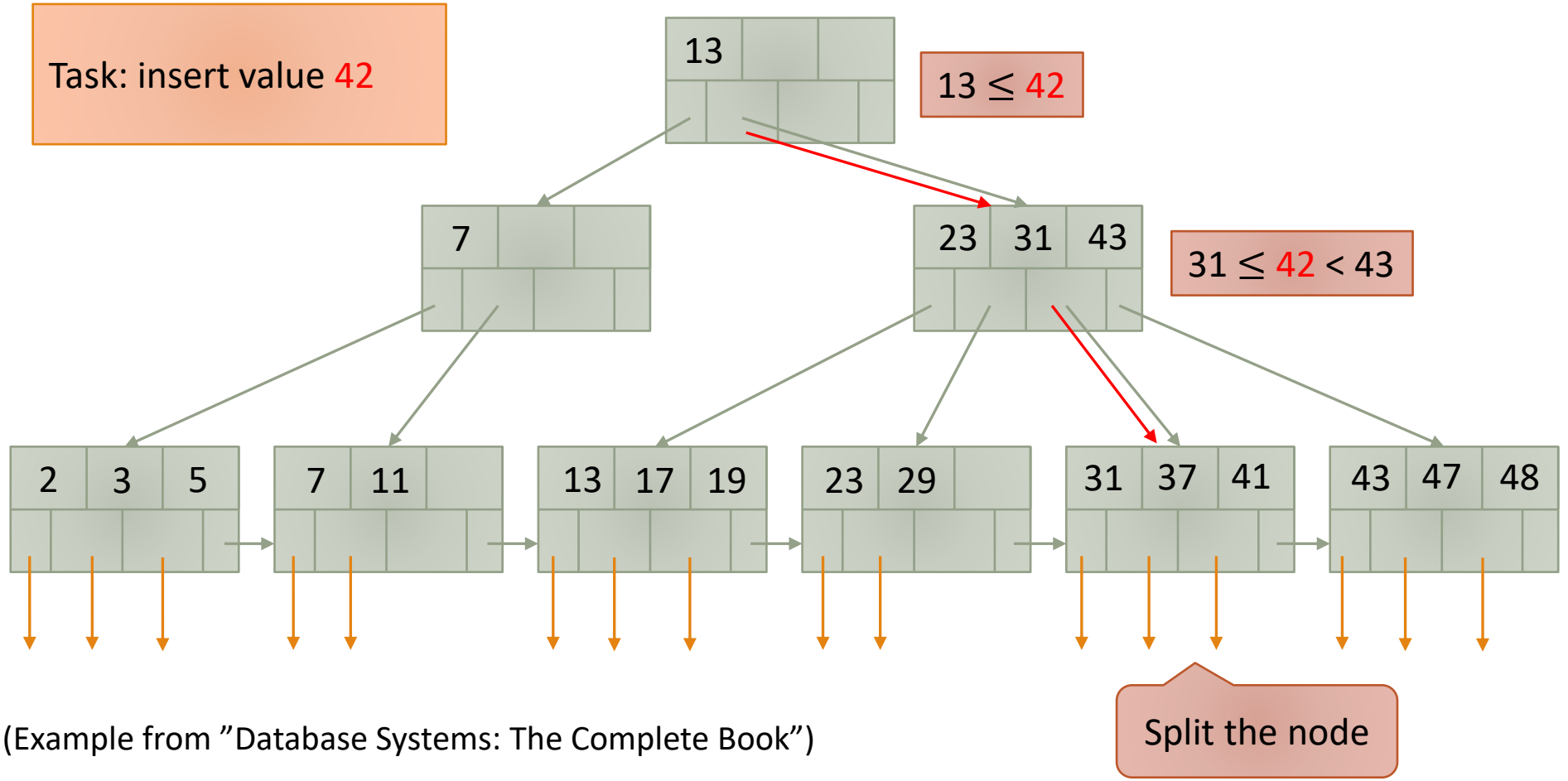


Insertions of Value/Pointer Pairs



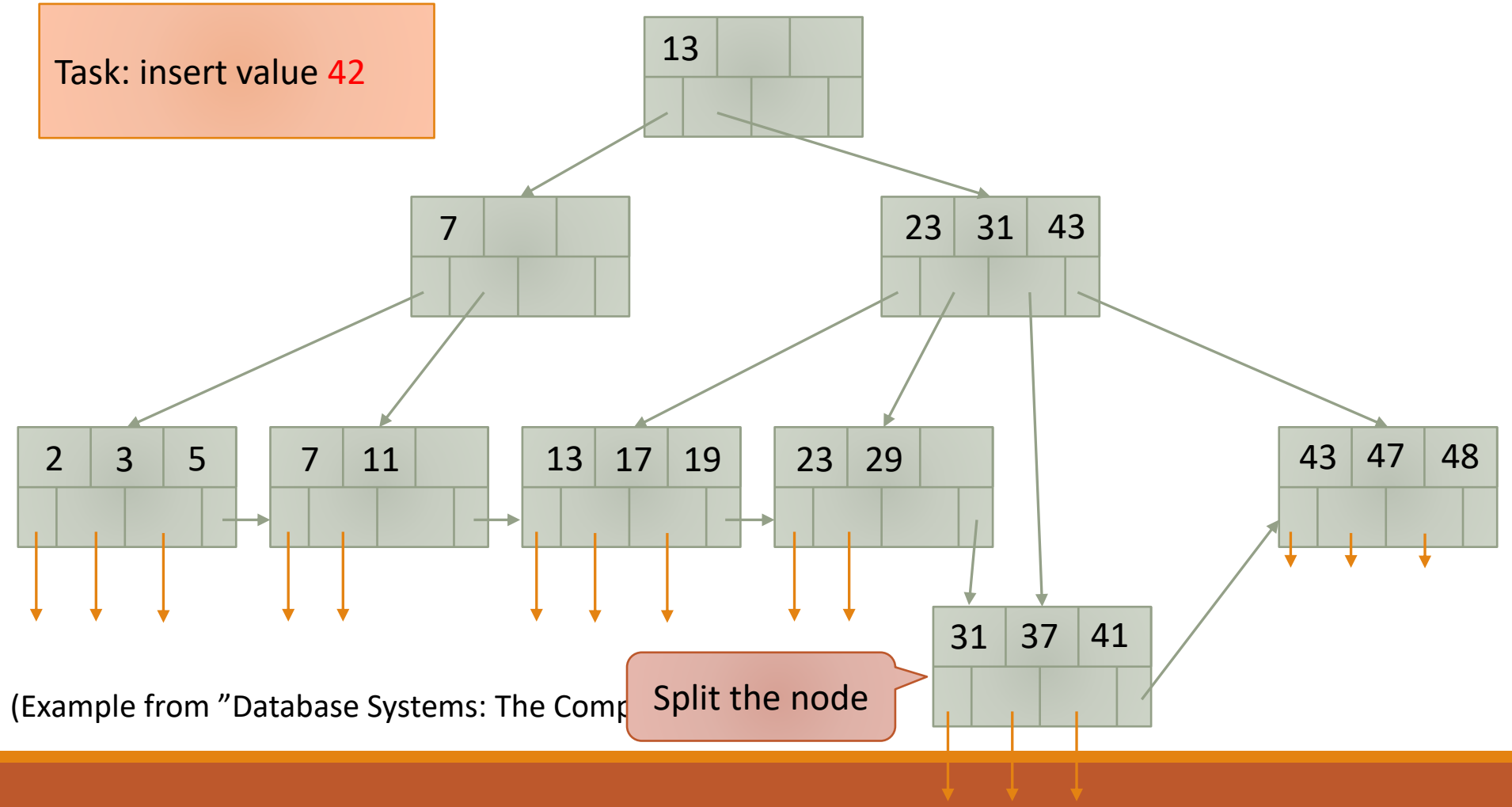
(Example from ”Database Systems: The Complete Book”)

Insertions of Value/Pointer Pairs



(Example from ”Database Systems: The Complete Book”)

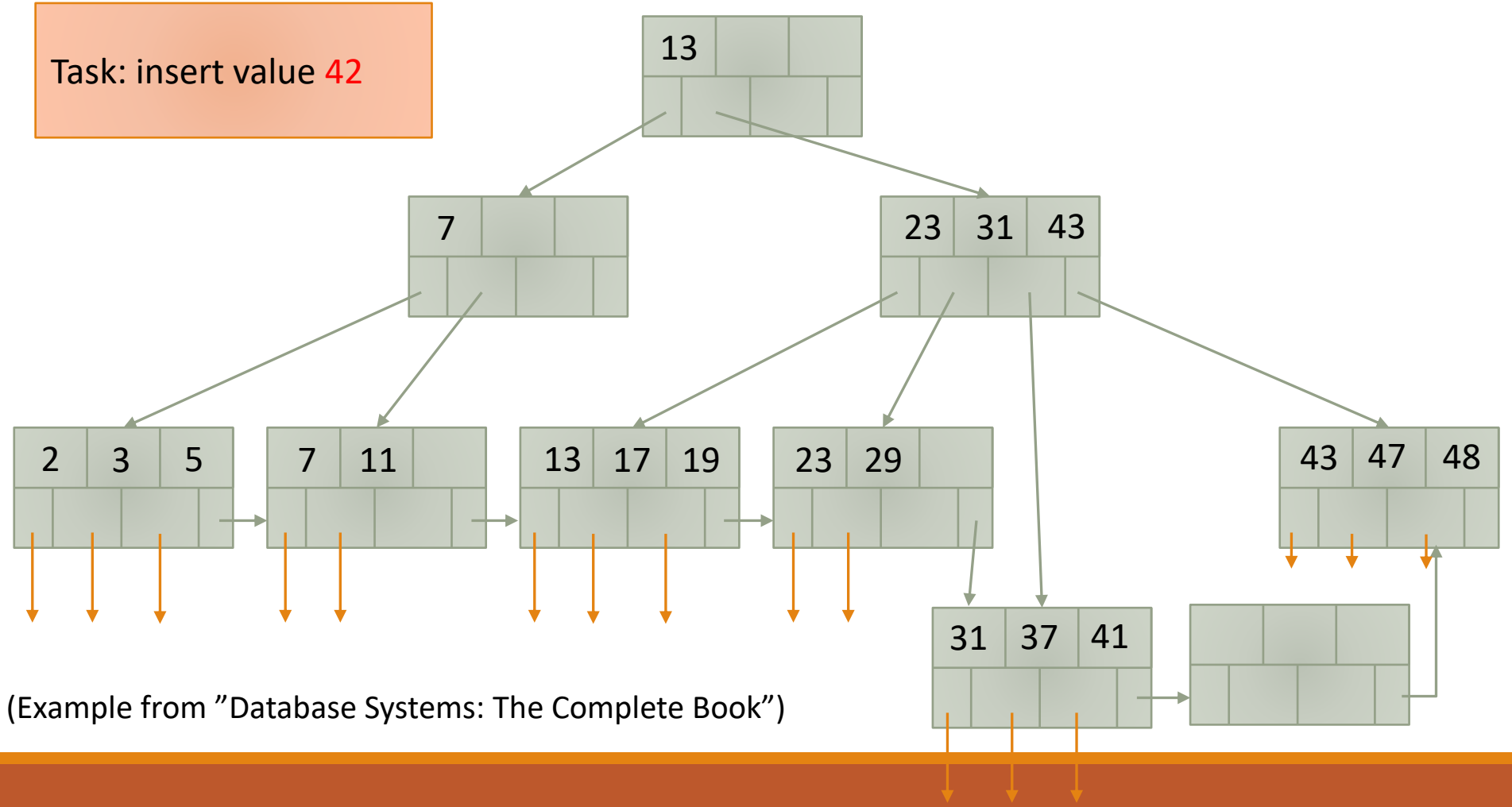
Insertions of Value/Pointer Pairs



(Example from ”Database Systems: The Comp

Insertions of Value/Pointer Pairs

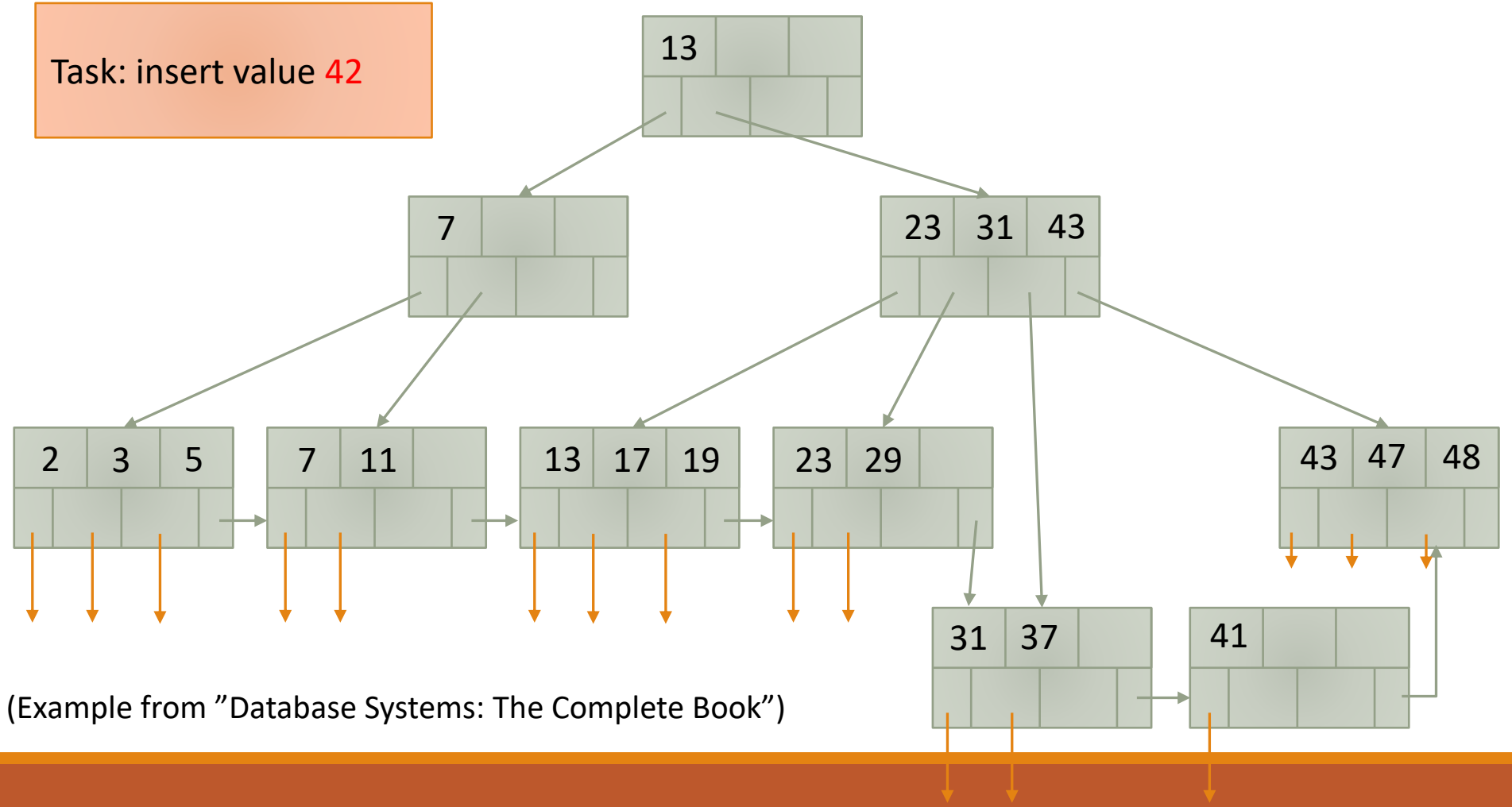
Task: insert value 42



(Example from ”Database Systems: The Complete Book”)

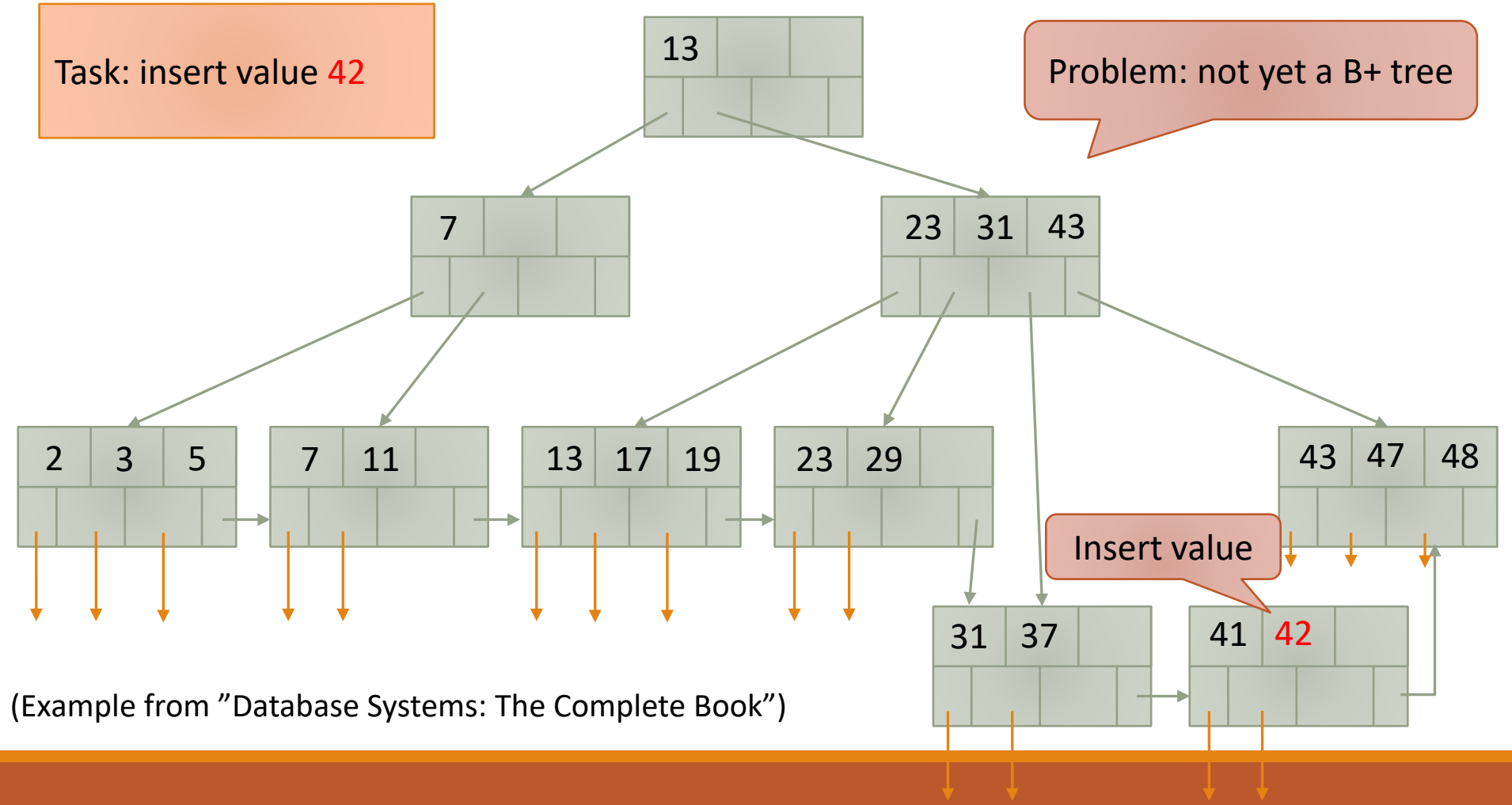
Insertions of Value/Pointer Pairs

Task: insert value 42



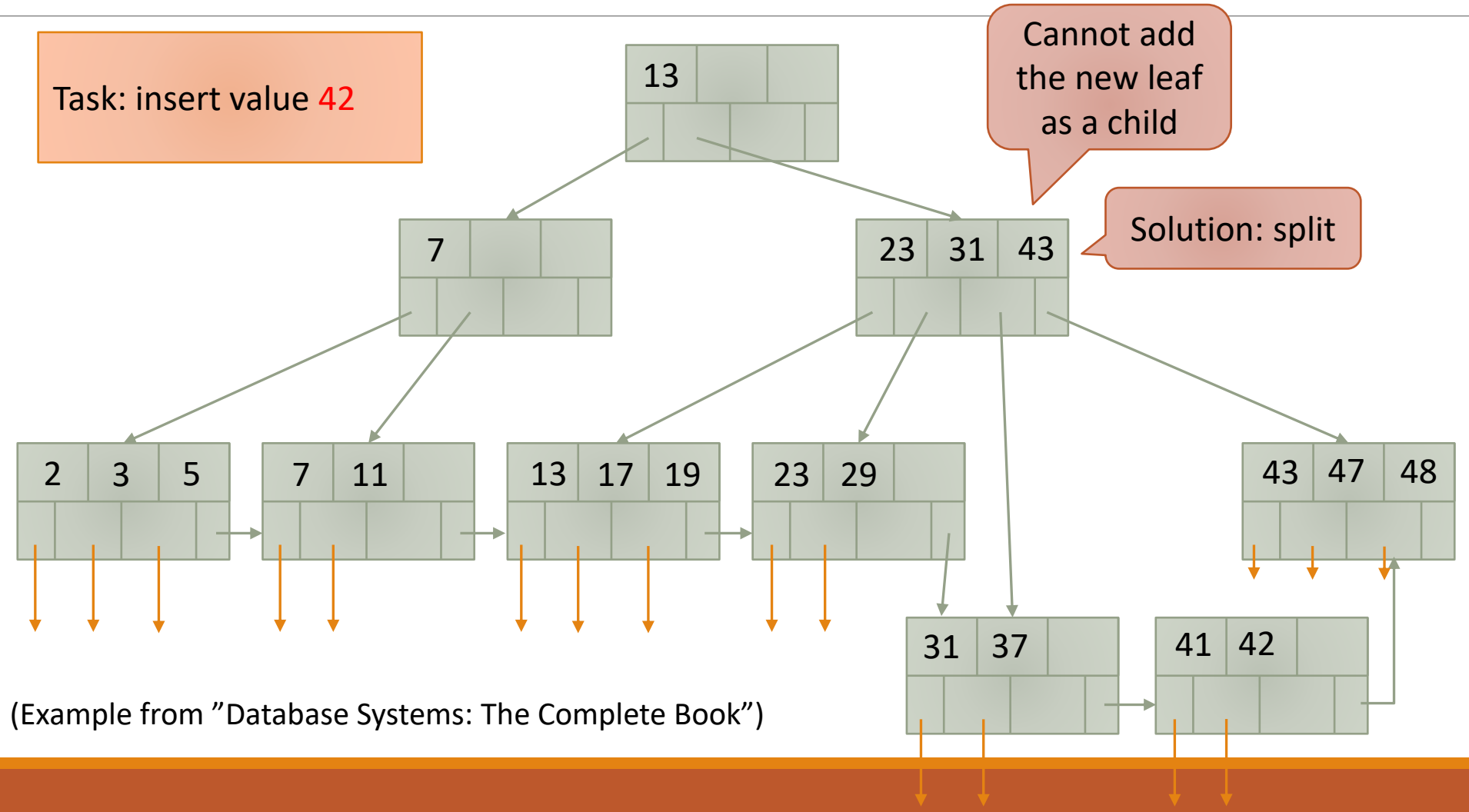
(Example from ”Database Systems: The Complete Book”)

Insertions of Value/Pointer Pairs



(Example from ”Database Systems: The Complete Book”)

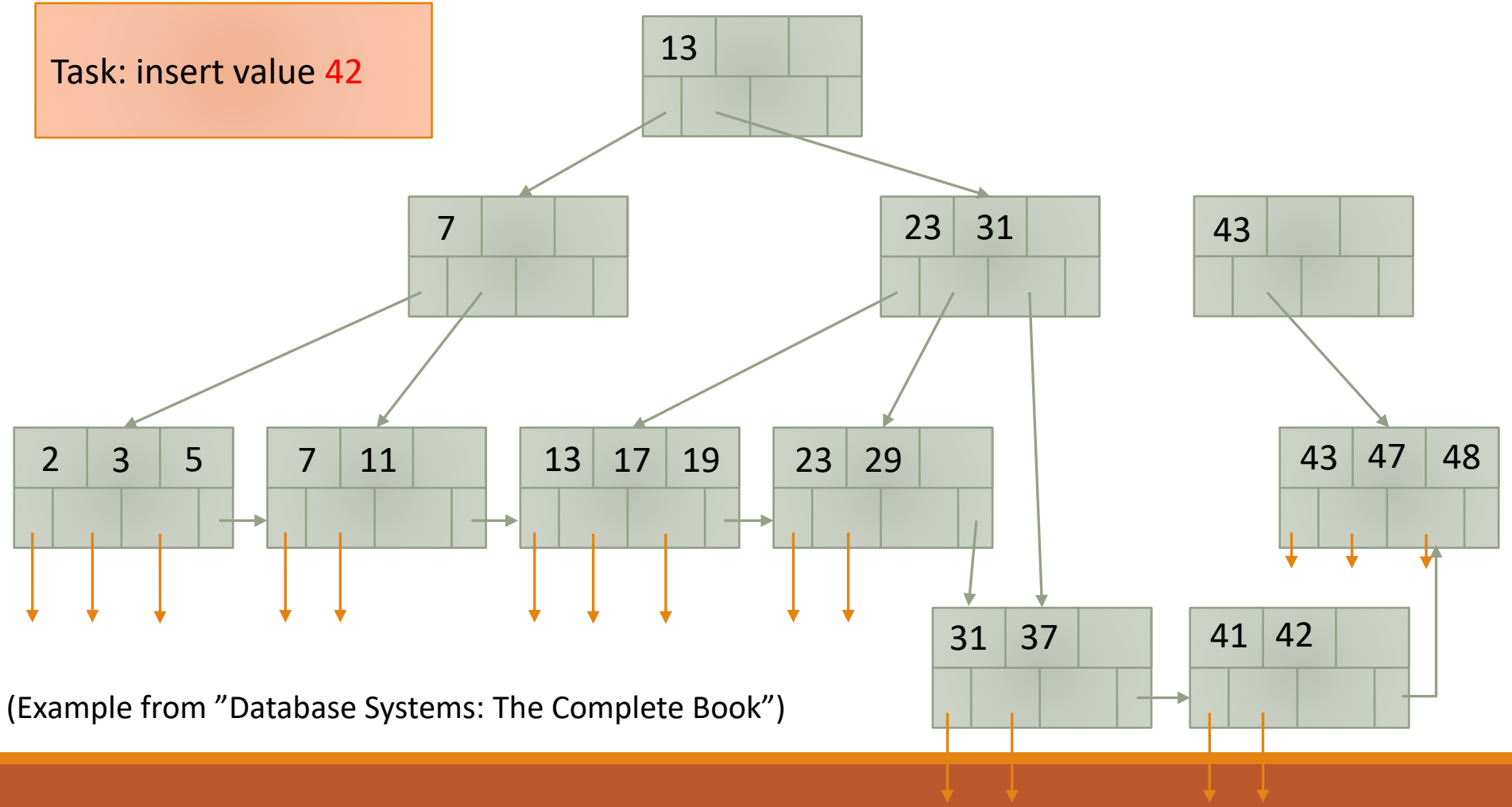
Insertions of Value/Pointer Pairs



(Example from ”Database Systems: The Complete Book”)

Insertions of Value/Pointer Pairs

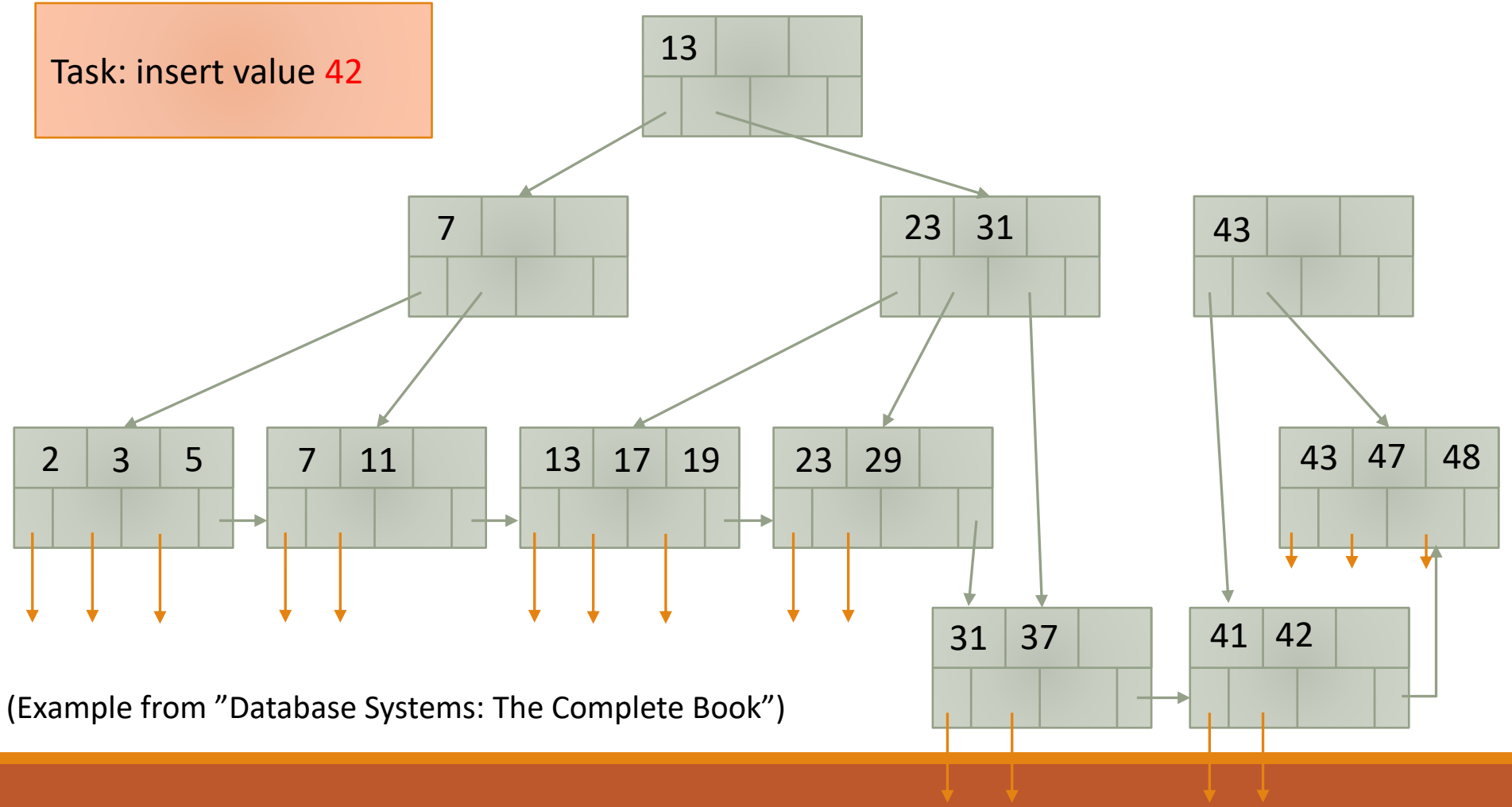
Task: insert value 42



(Example from ”Database Systems: The Complete Book”)

Insertions of Value/Pointer Pairs

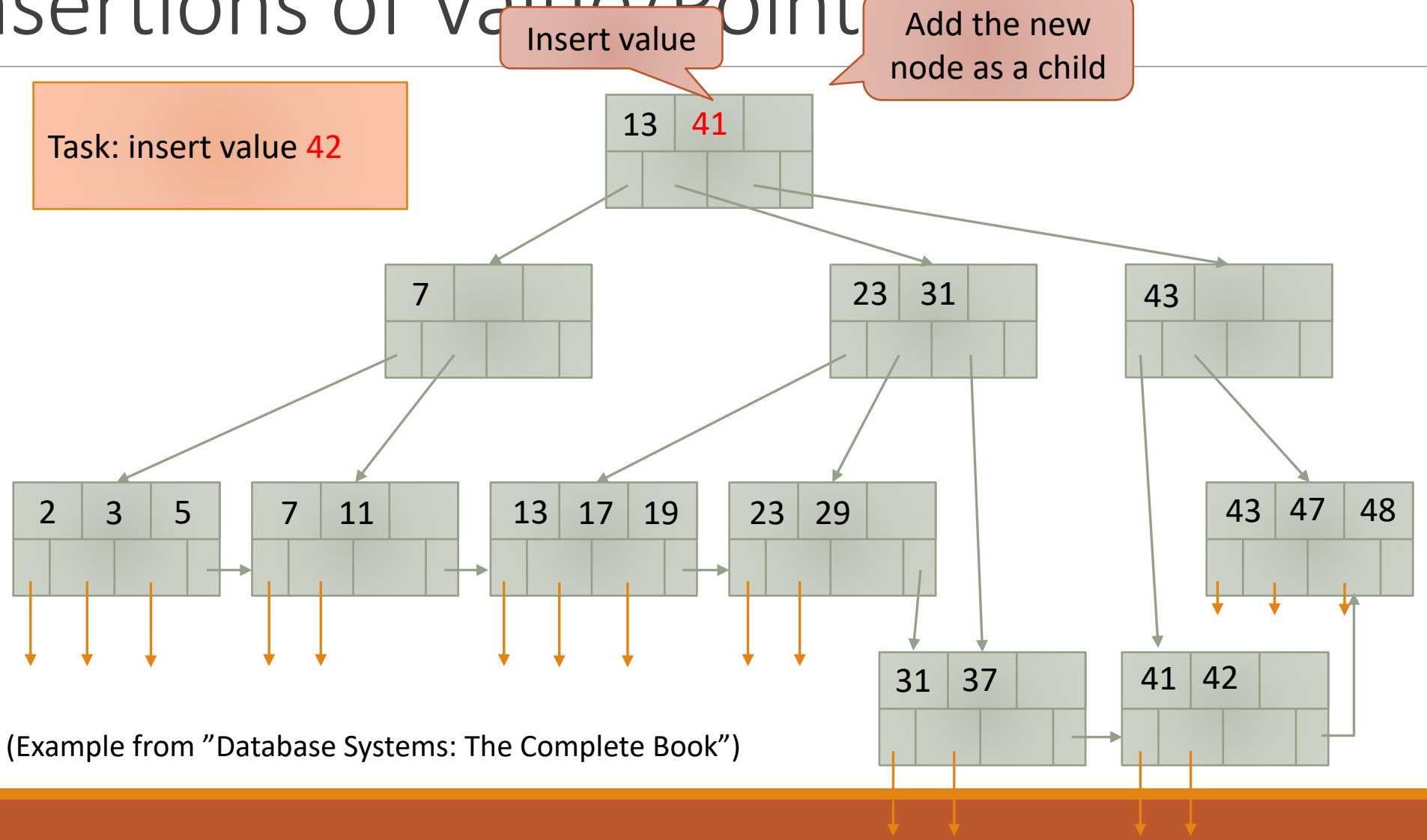
Task: insert value 42



(Example from ”Database Systems: The Complete Book”)

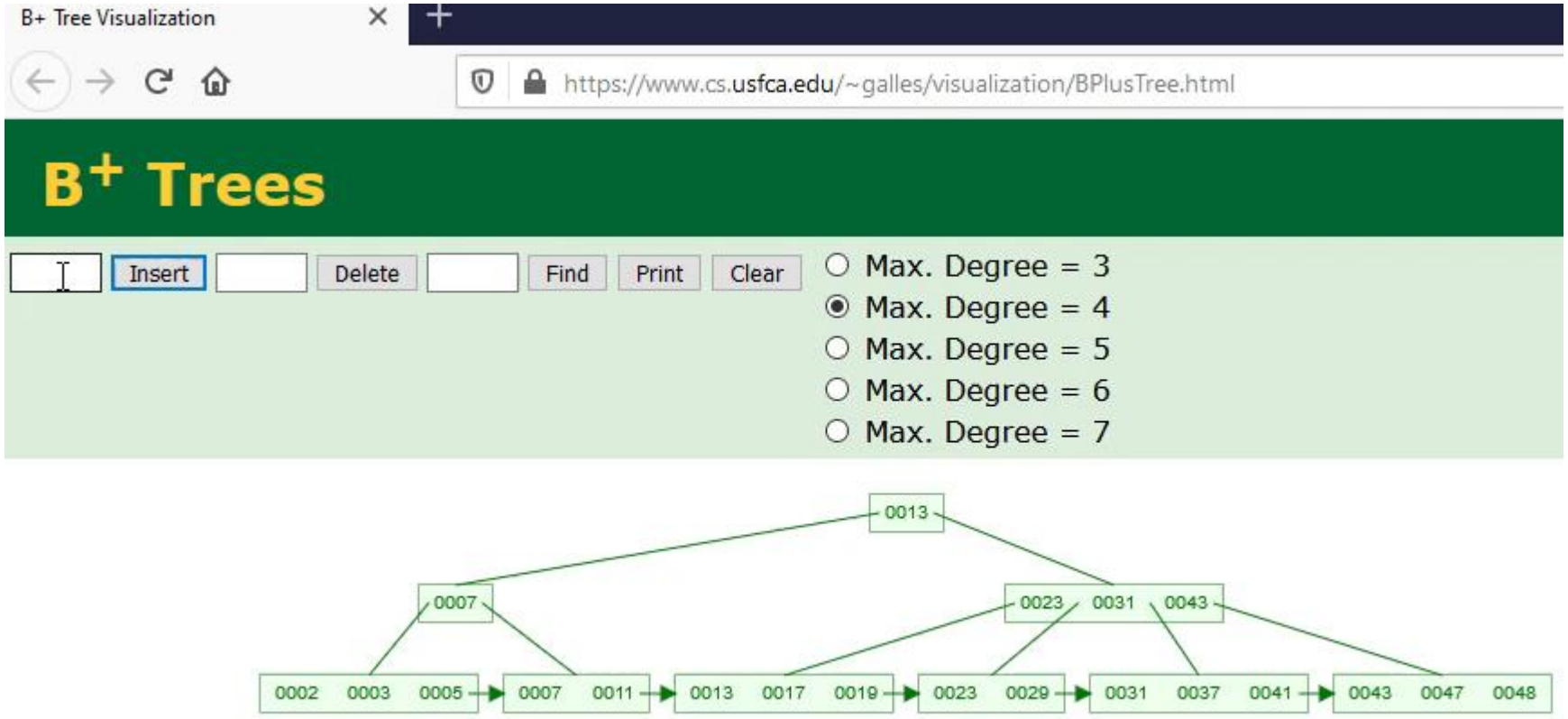
Insertions of Value/Pointer Pairs

Task: insert value 42



(Example from ”Database Systems: The Complete Book”)

Online version of inserting 42



Insertion: Summary

Goal: insert a new value/pointer pair

Procedure:

- Find the leaf that should contain the value
- If the leaf is not full, insert the key value pair at a suitable location
- If the leaf is full:
  - Split the leaf to make space for the new value/pointer pair and move half of the pointers to the new node
  - Insert the value/pointer pair
  - Connect the leaf to a suitable parent node (which might incur the creation of a new node etc.)

The B+ tree **remains balanced**!

Time for a disk operation

Running time: $O(h \times \log_2 n)$ "real" running time $O(h \times D)$

Height of the B+ tree

Summary

The video provided an overview over the idea of B+-trees and how to search and insert into them

- Slide 16 has a summary of lookups
- Slide 27 has a summary of insertions