Lecture#12

Data Structures

Dr. Abu Nowshed Chy

Department of Computer Science and Engineering

University of Chittagong

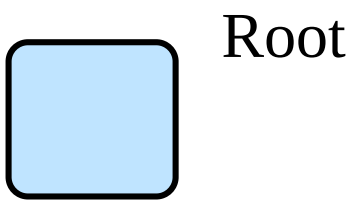
February 26, 2025

Faculty Profile

Tree

Complete Binary Tree

- Every level except bottom is complete.
- On the bottom, nodes are placed as left as possible.

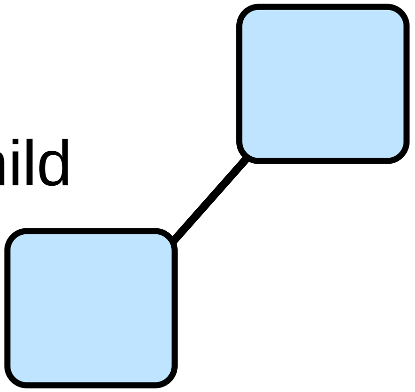


When a complete binary tree is built, its first node must be the root.

Complete Binary Tree

Complete binary tree.

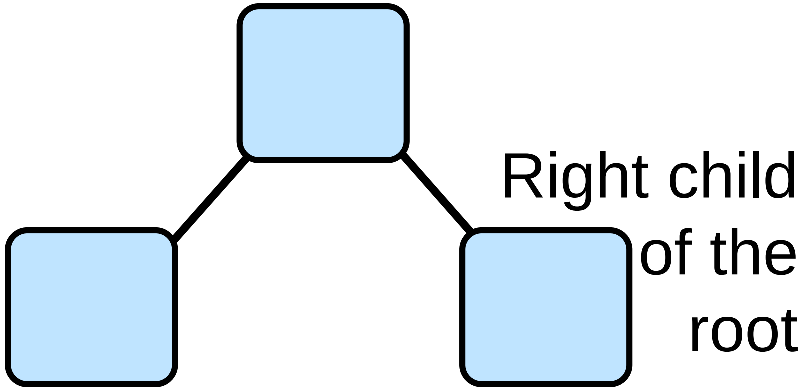
Left child of the root



The second node is always the left child of the root.

Complete Binary Tree

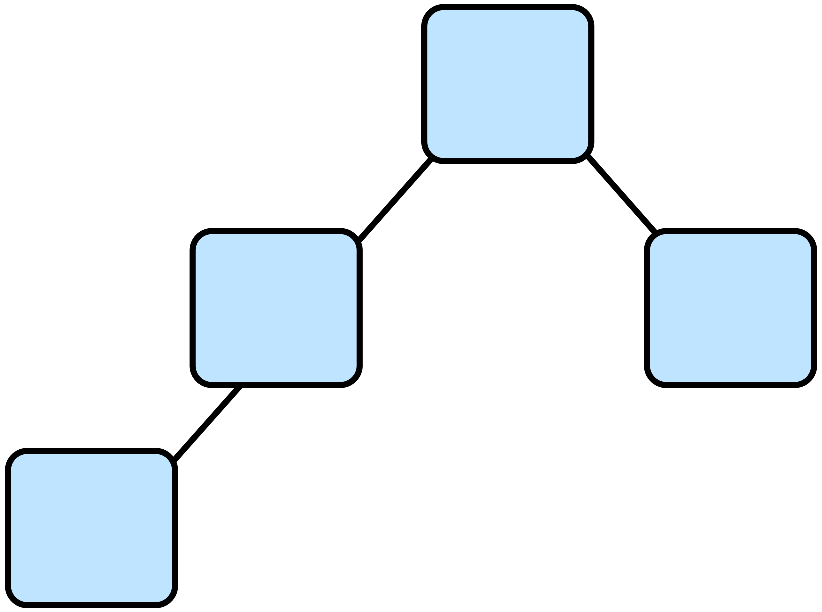
Complete binary tree.



The third node is always the right child of the root.

Complete Binary Tree

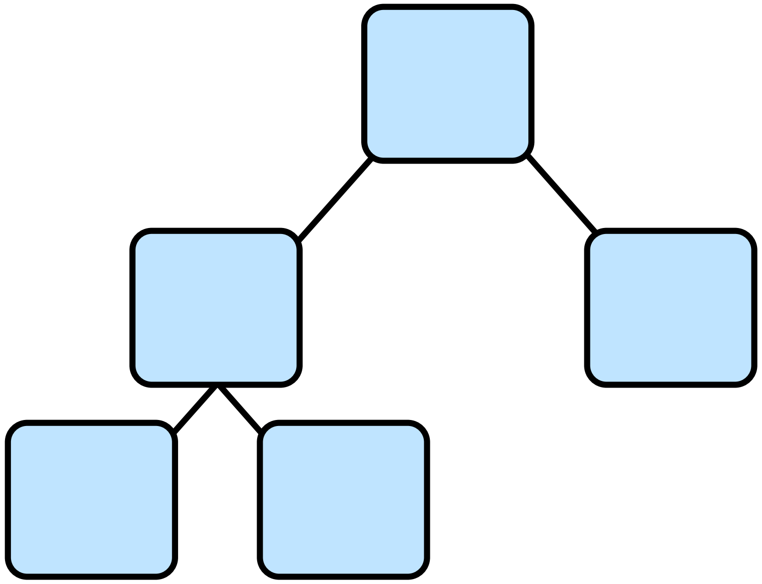
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

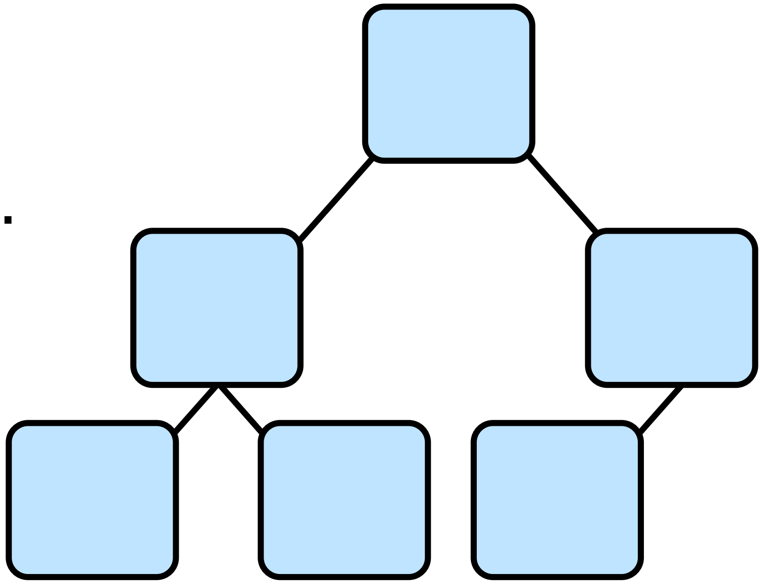
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

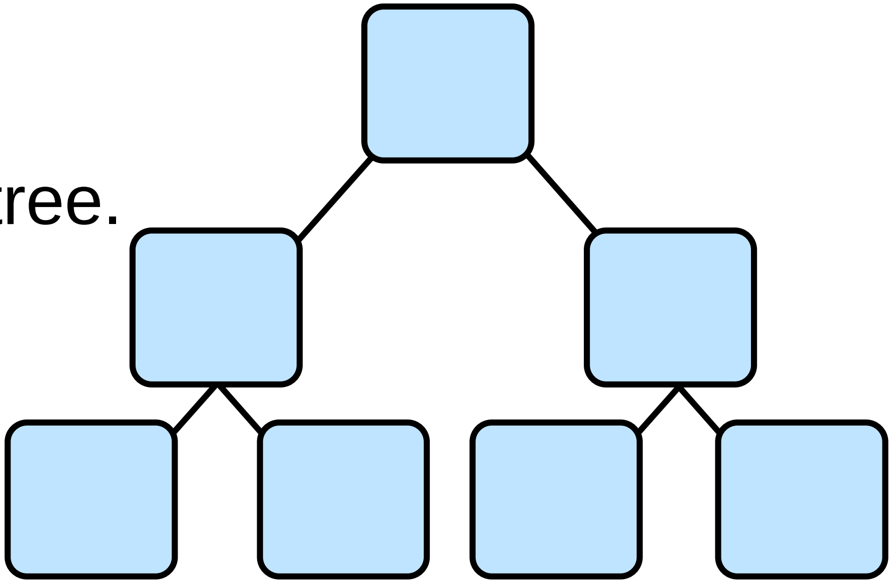
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

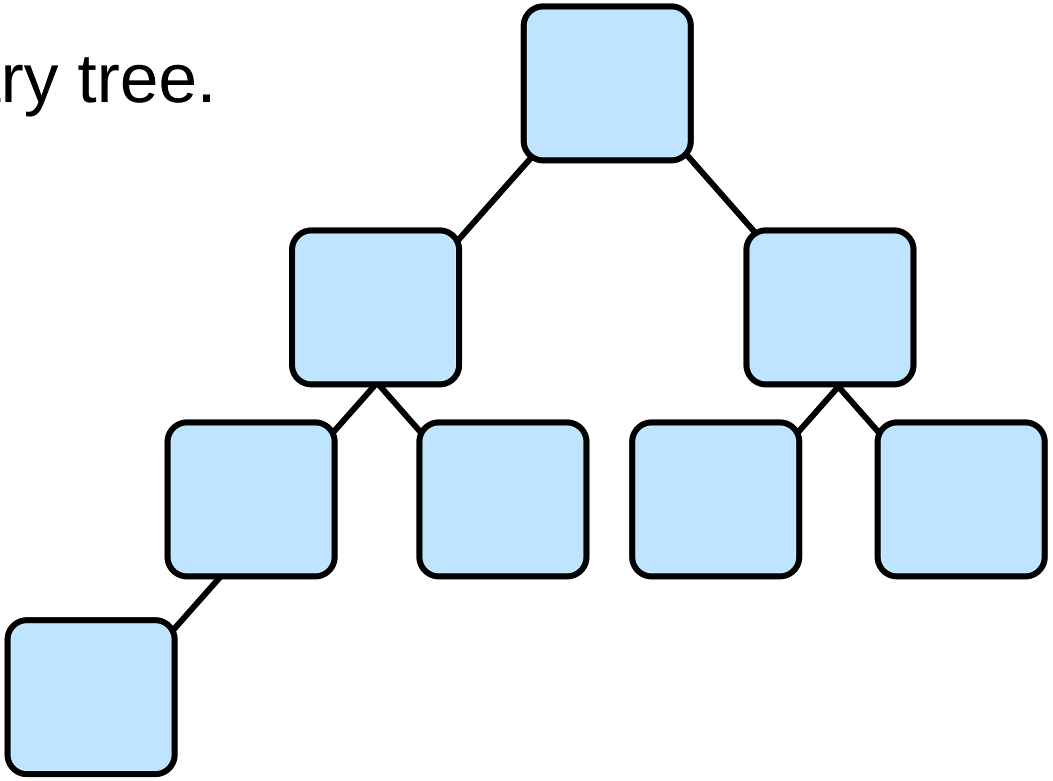
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

Complete binary tree.



Max Heap

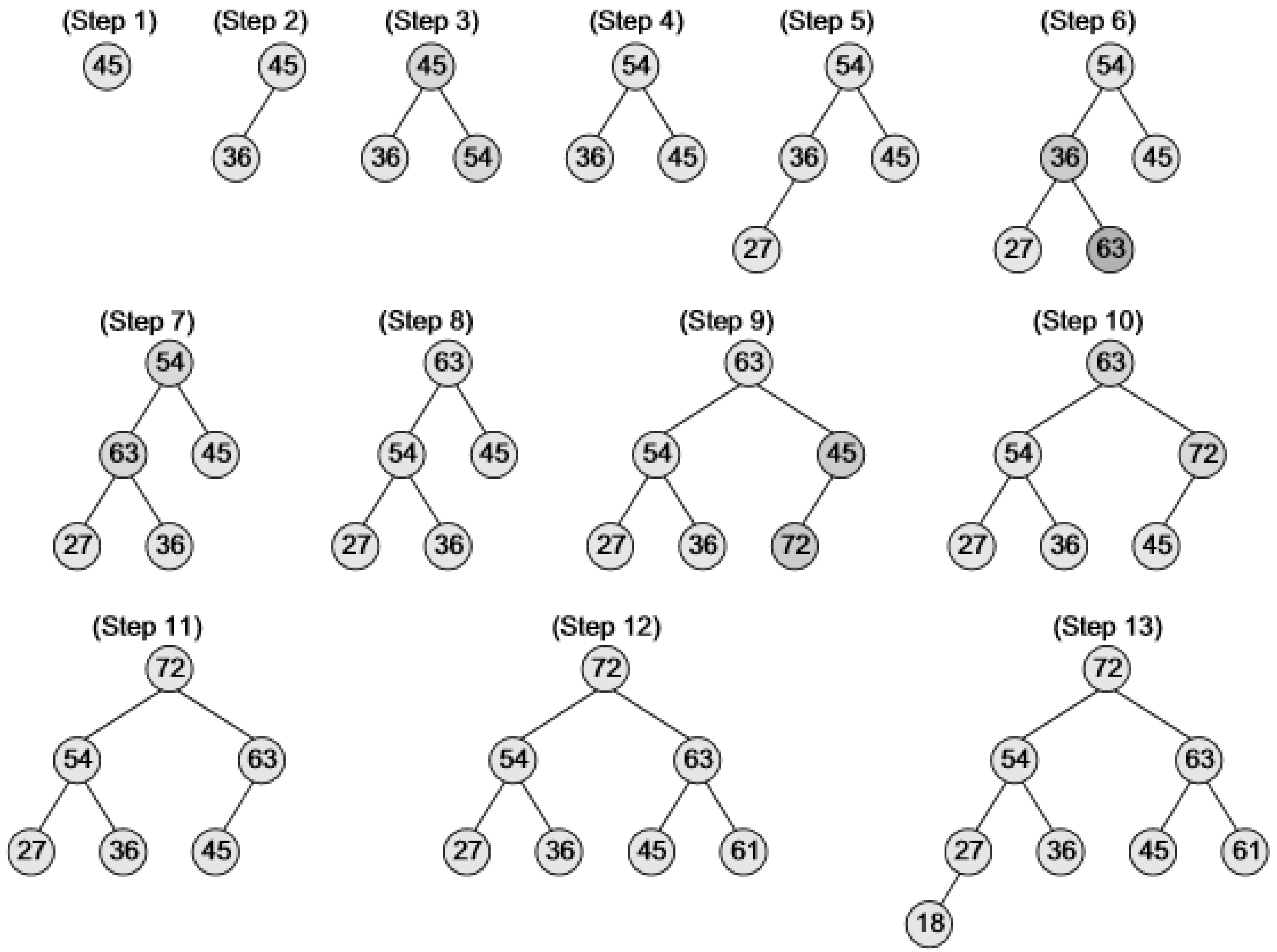
A heap is a certain kind of nearly **complete binary tree**.

Build a max heap H from the given set of numbers:

45, 36, 54, 27, 63, 72, 61,and 18

Max Heap

45, 36, 54, 27, 63, 72, 61, 18

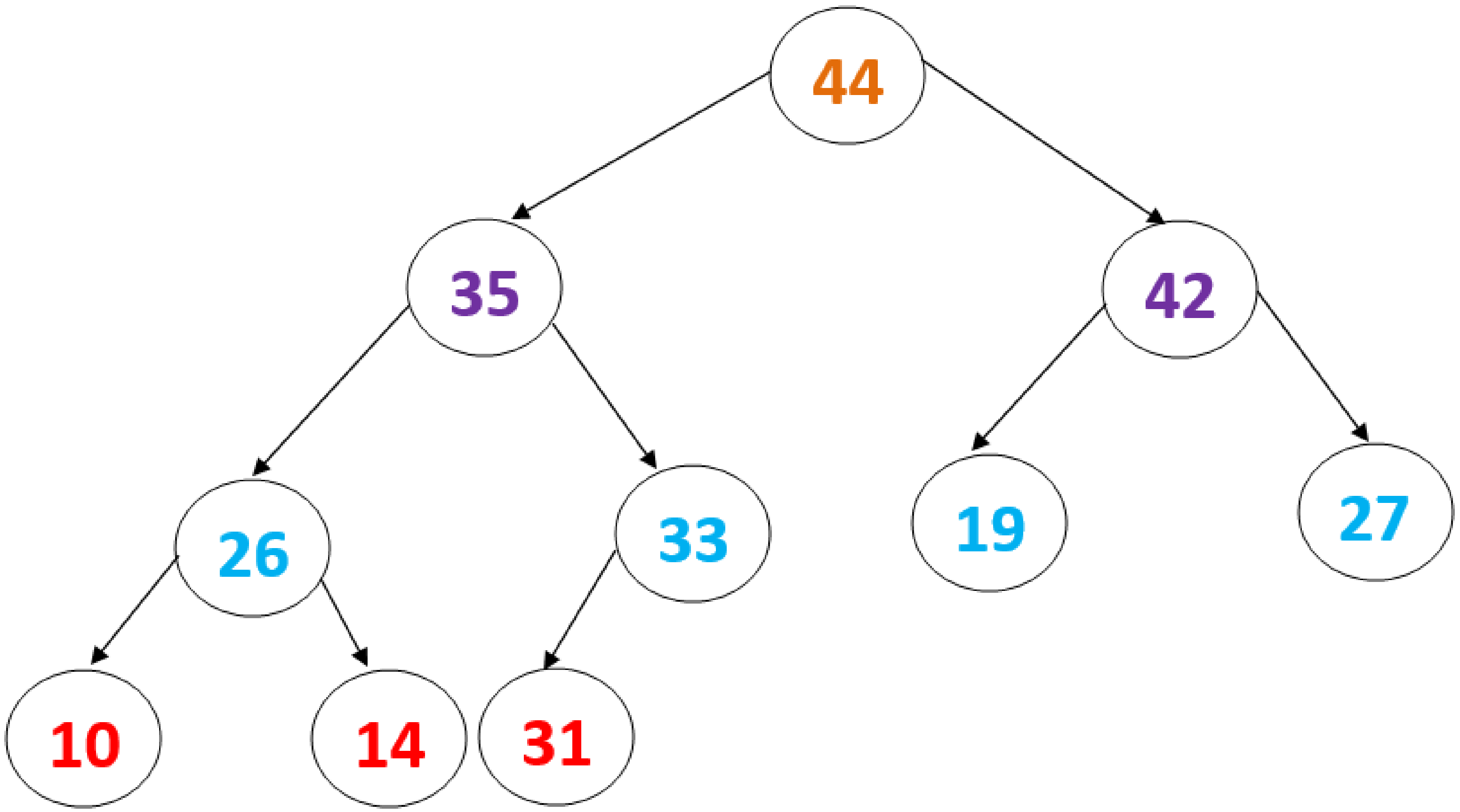


Max Heap

Build a max heap H from the given set of numbers:

44, 35, 42, 26, 33, 19, 27, 10, 14, and 31

Max Heap

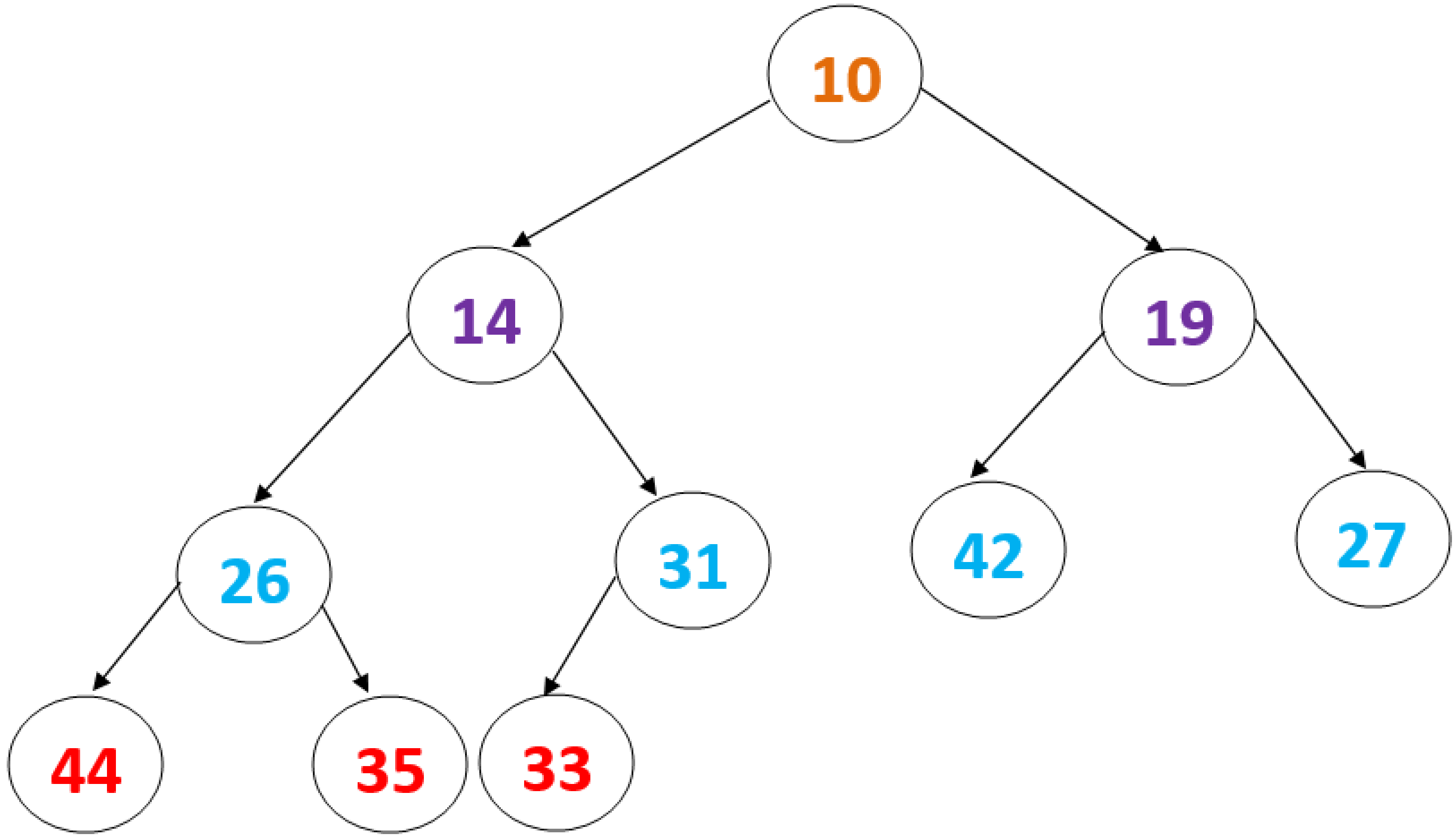


Min Heap

Build a min heap H from the given set of numbers:

10, 14, 19, 26, 31, 42, 27, 44, 35, and 33

Min Heap



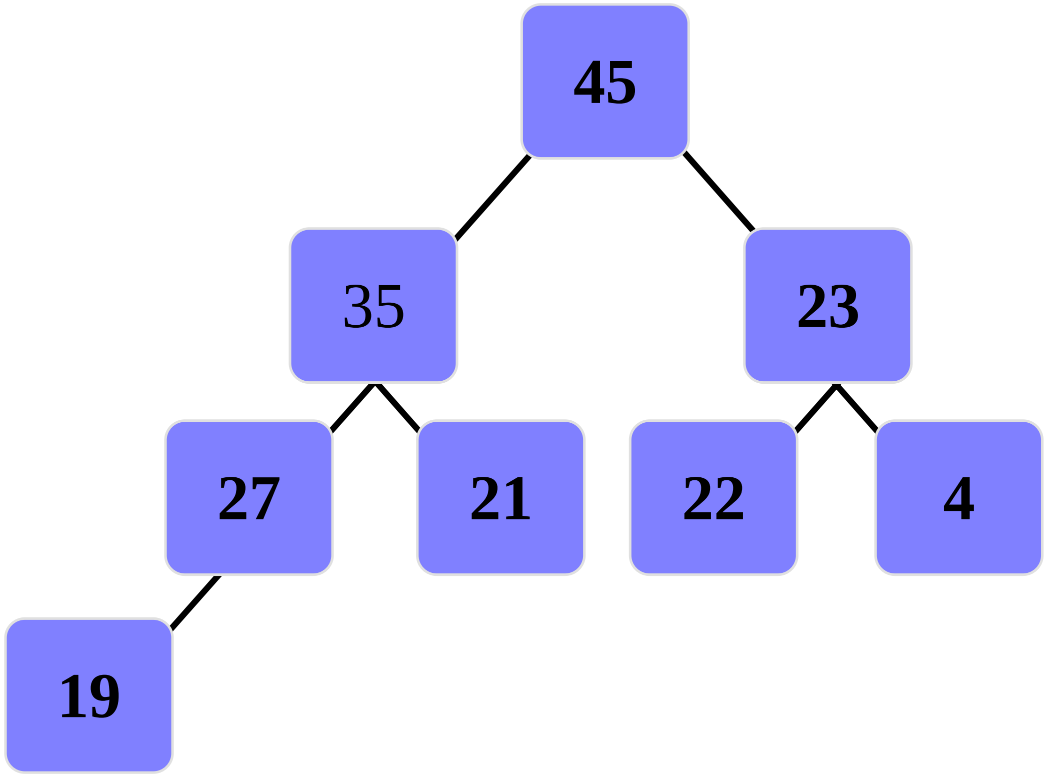
Heap

Build a max and min heap H from the given set of numbers:

85, 31, 21, 30, 51, 10, 06, 22, 25, 37, 41, 22, 87, 09

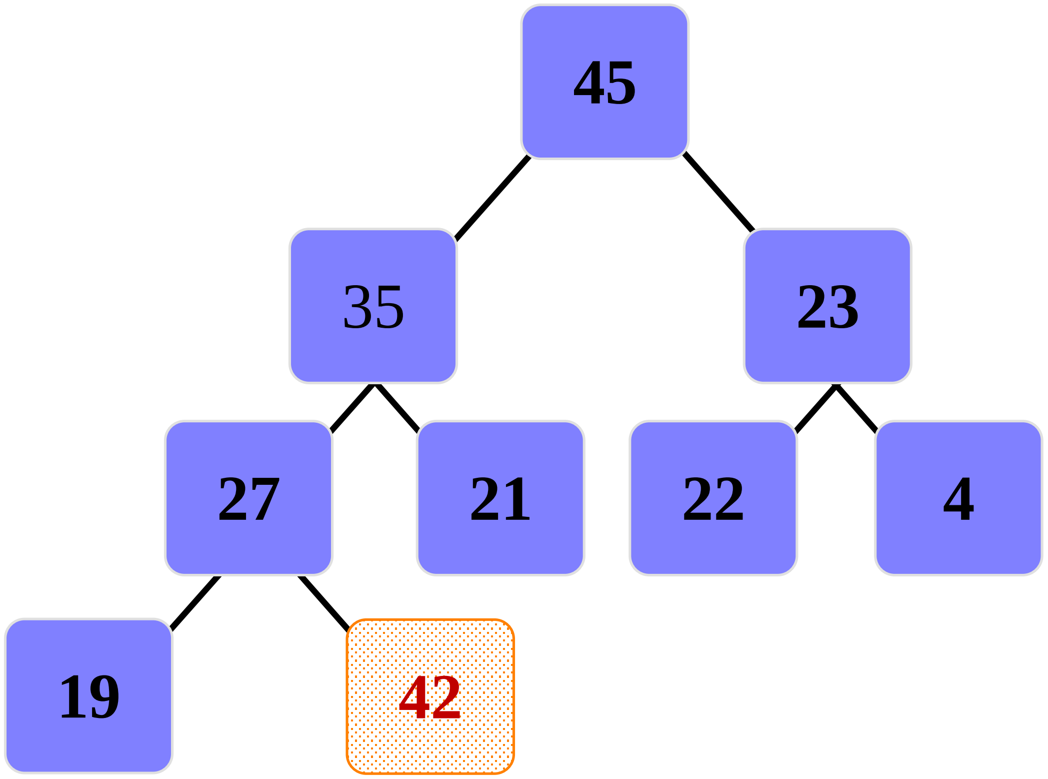
Insert New Node into Heap

Insert 42



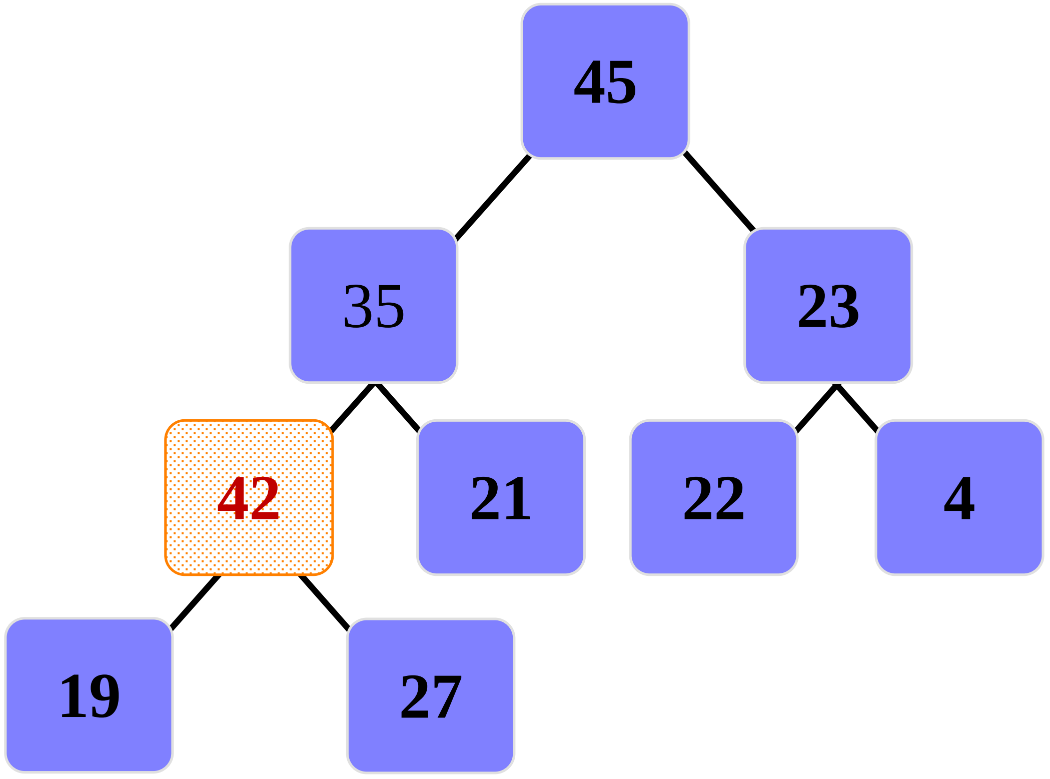
Insert New Node into Heap

Insert 42



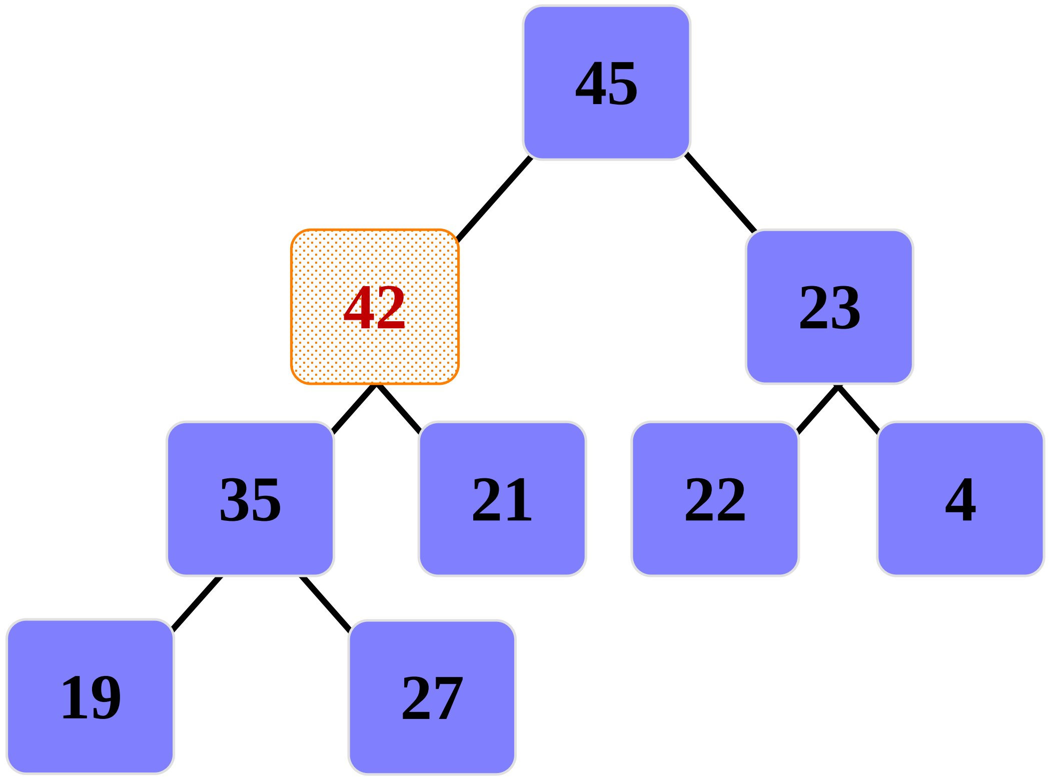
Insert New Node into Heap

Insert 42



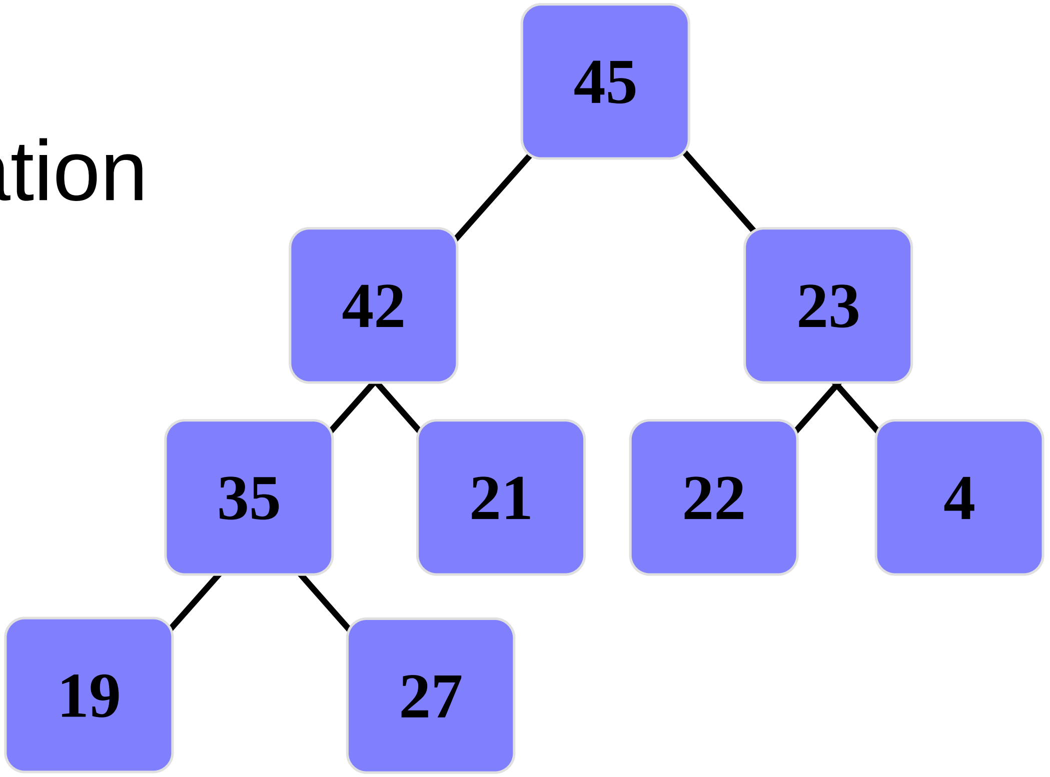
Insert New Node into Heap

Insert 42



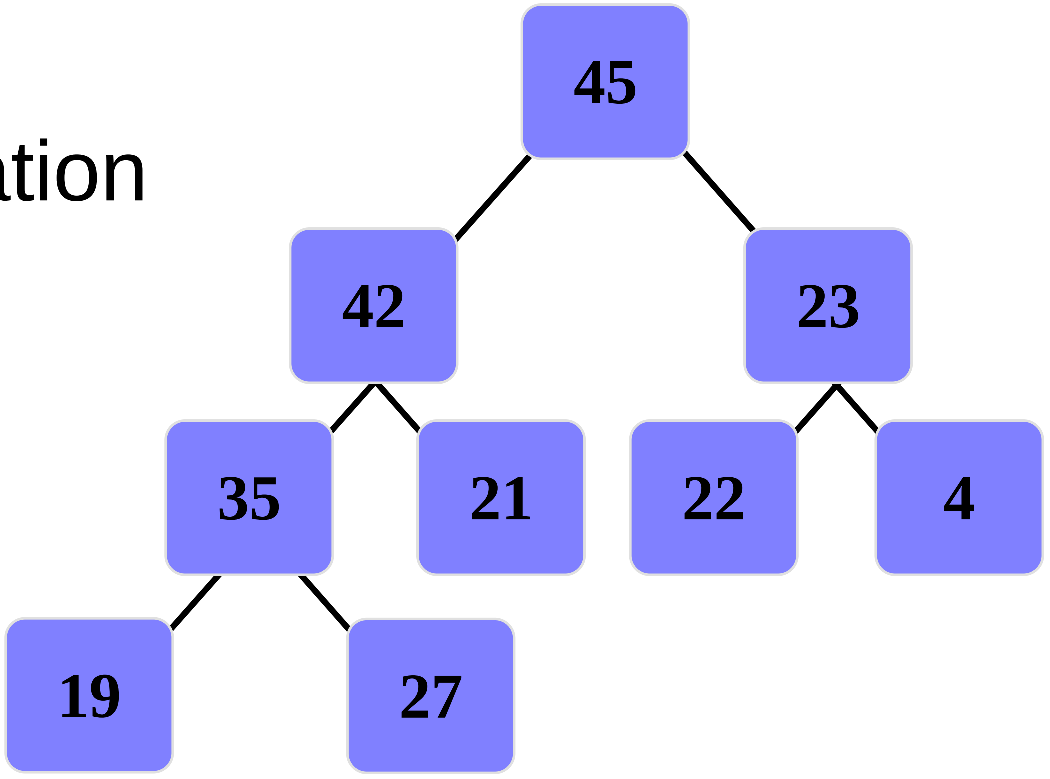
Insert New Node into Heap

Reheapification Upward



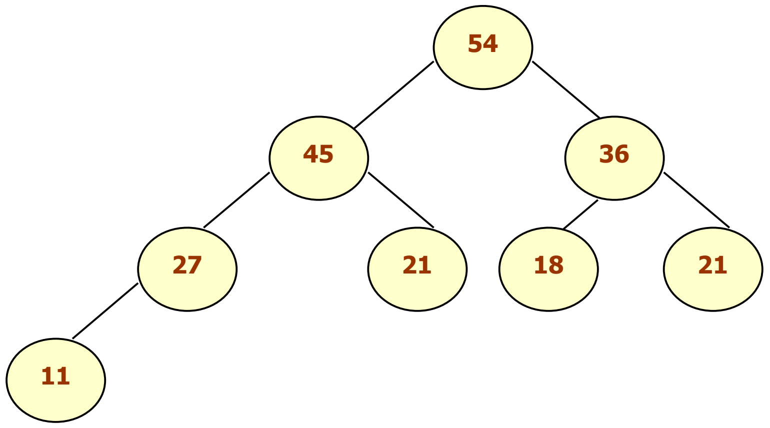
Insert New Node into Heap

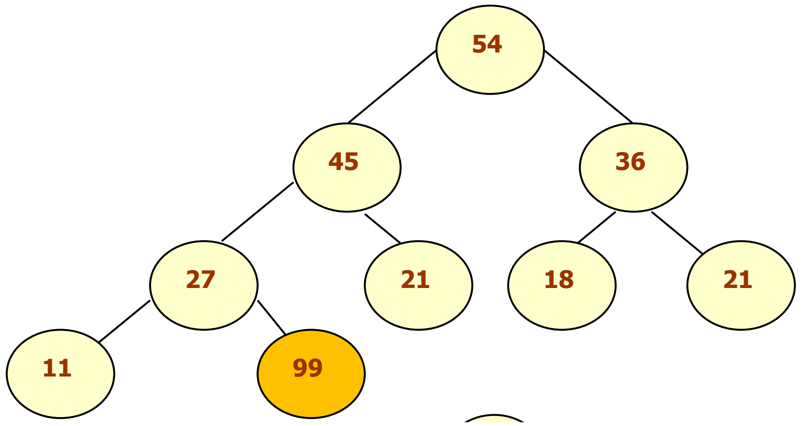
Reheapification
Upward

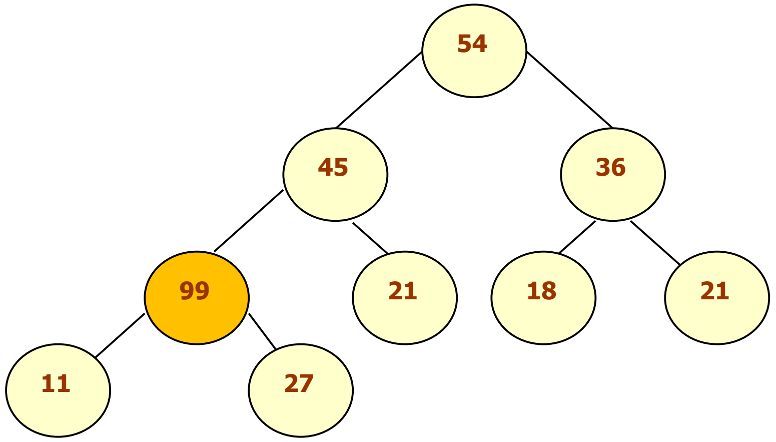


Insert New Node into Heap

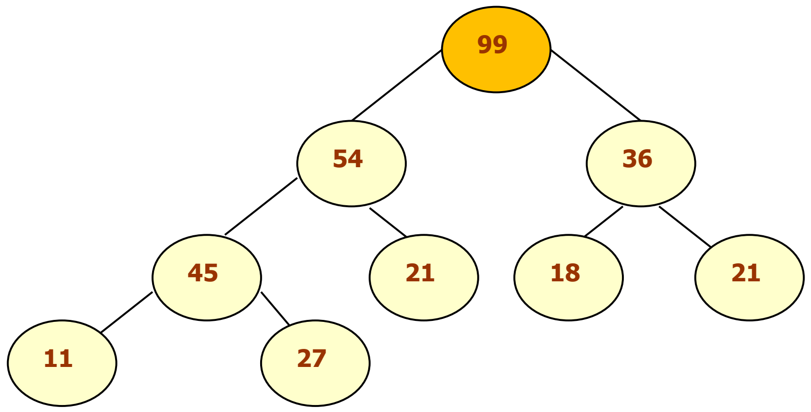
Consider the heap given below and insert 99 in it





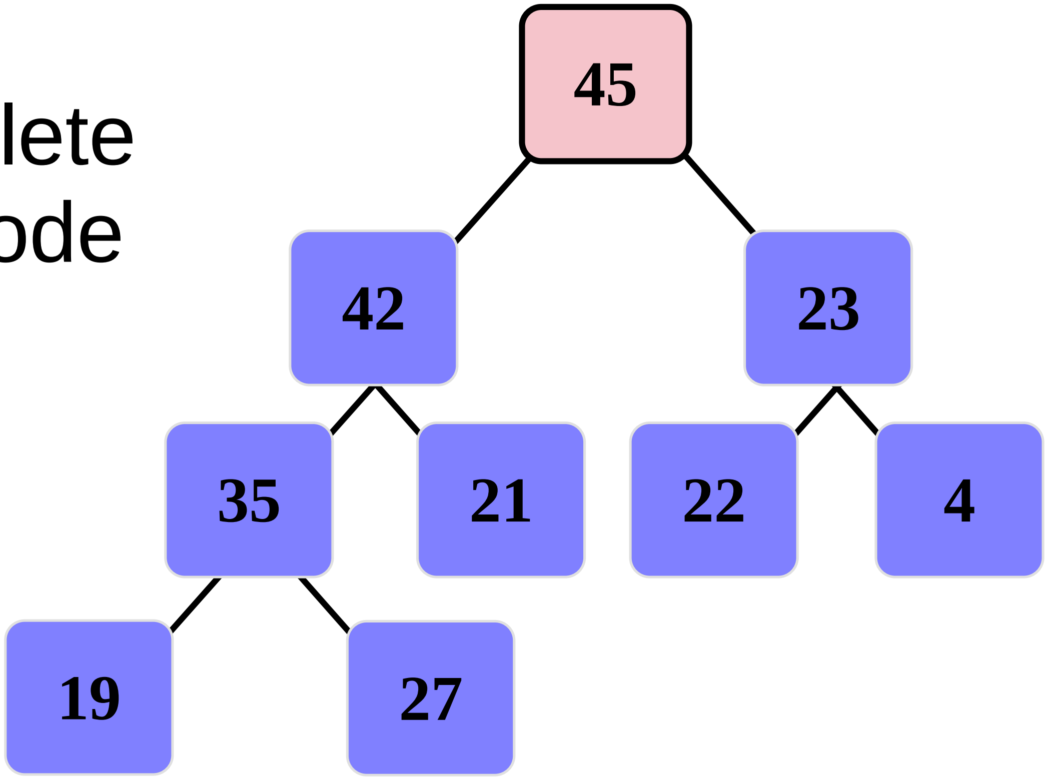


Insert New Node into Heap



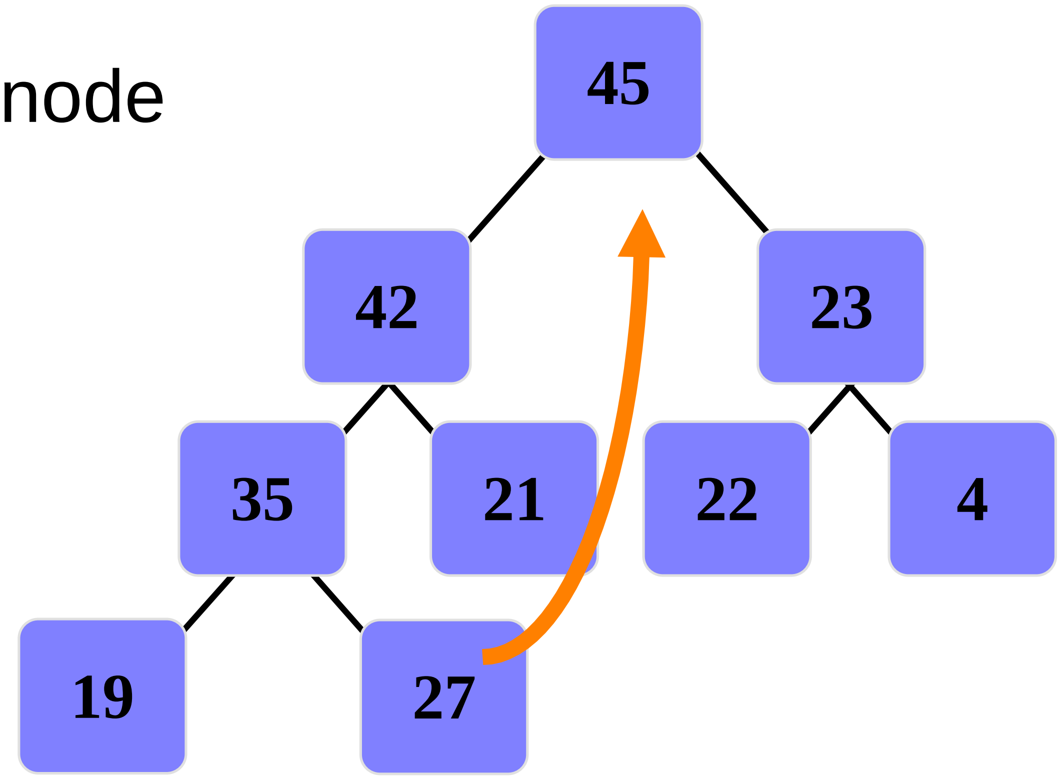
Delete a Node from Heap

Always delete the root node



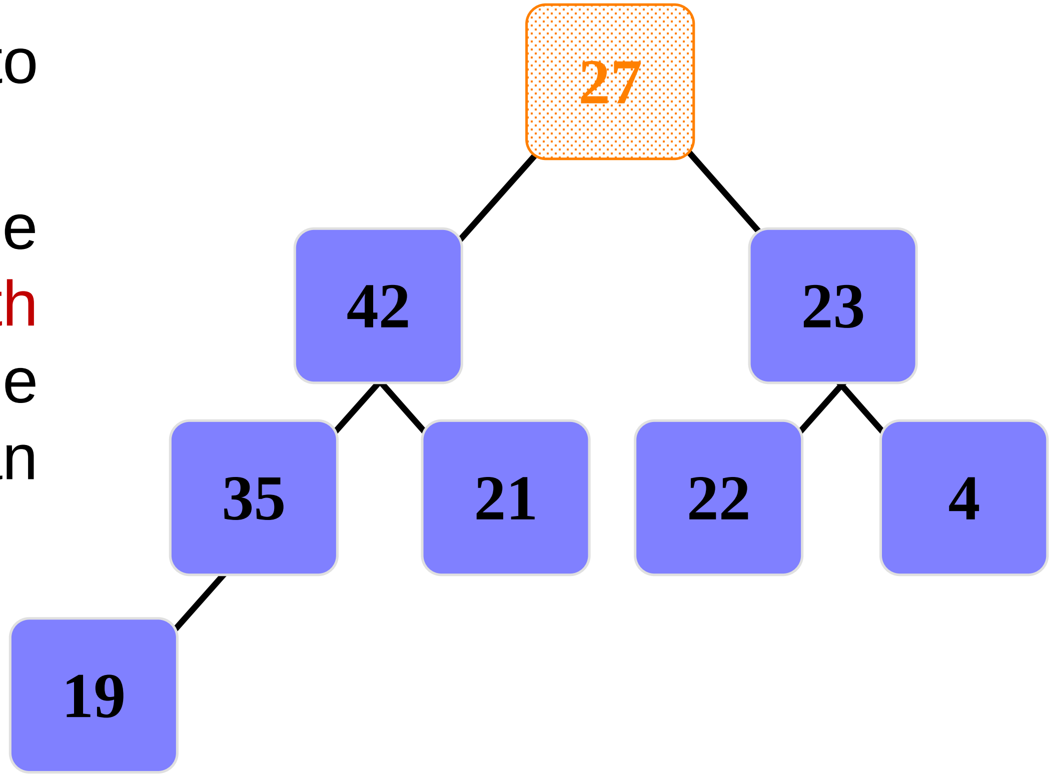
Delete a Node from Heap

Move the last node onto the root.

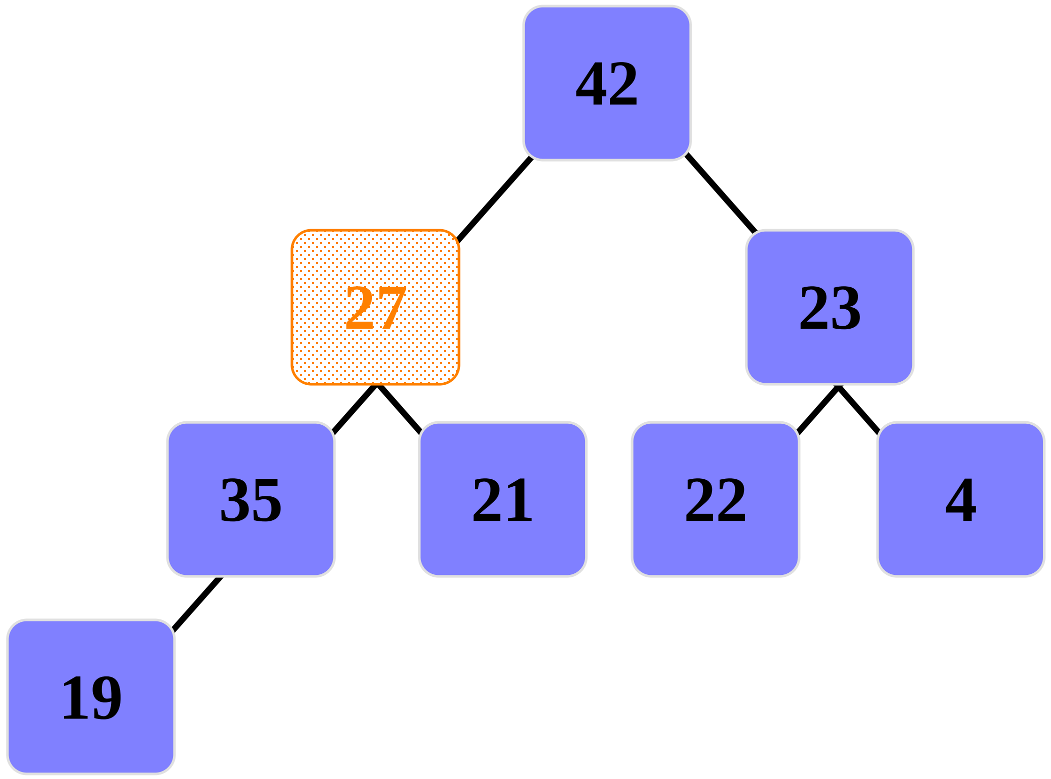


Delete a Node from Heap

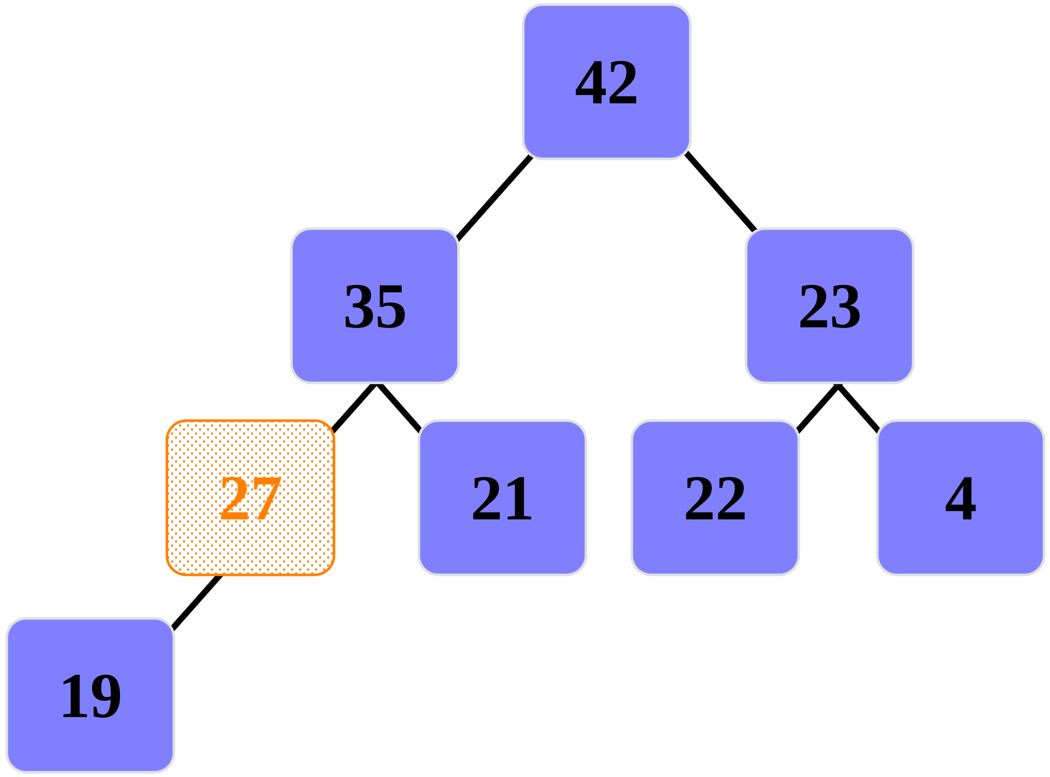
- Move the last node onto the root.
- Push the out-of-place node downward, swapping with its larger child until the new node reaches an acceptable location.



Delete a Node from Heap

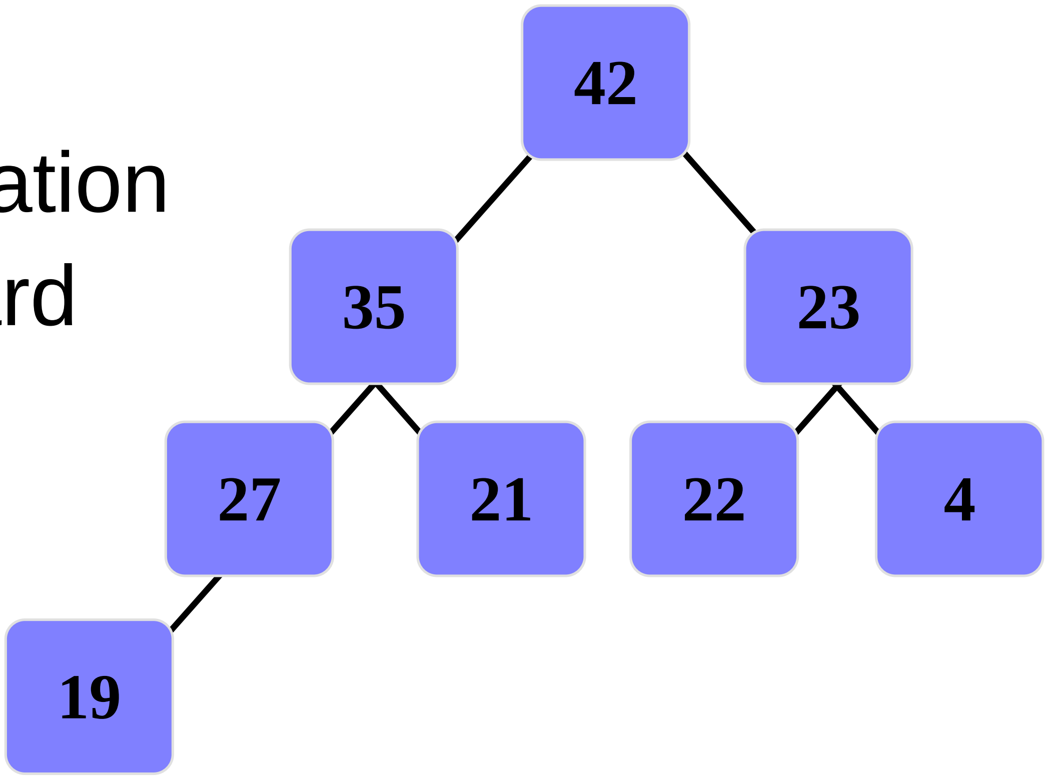


Delete a Node from Heap



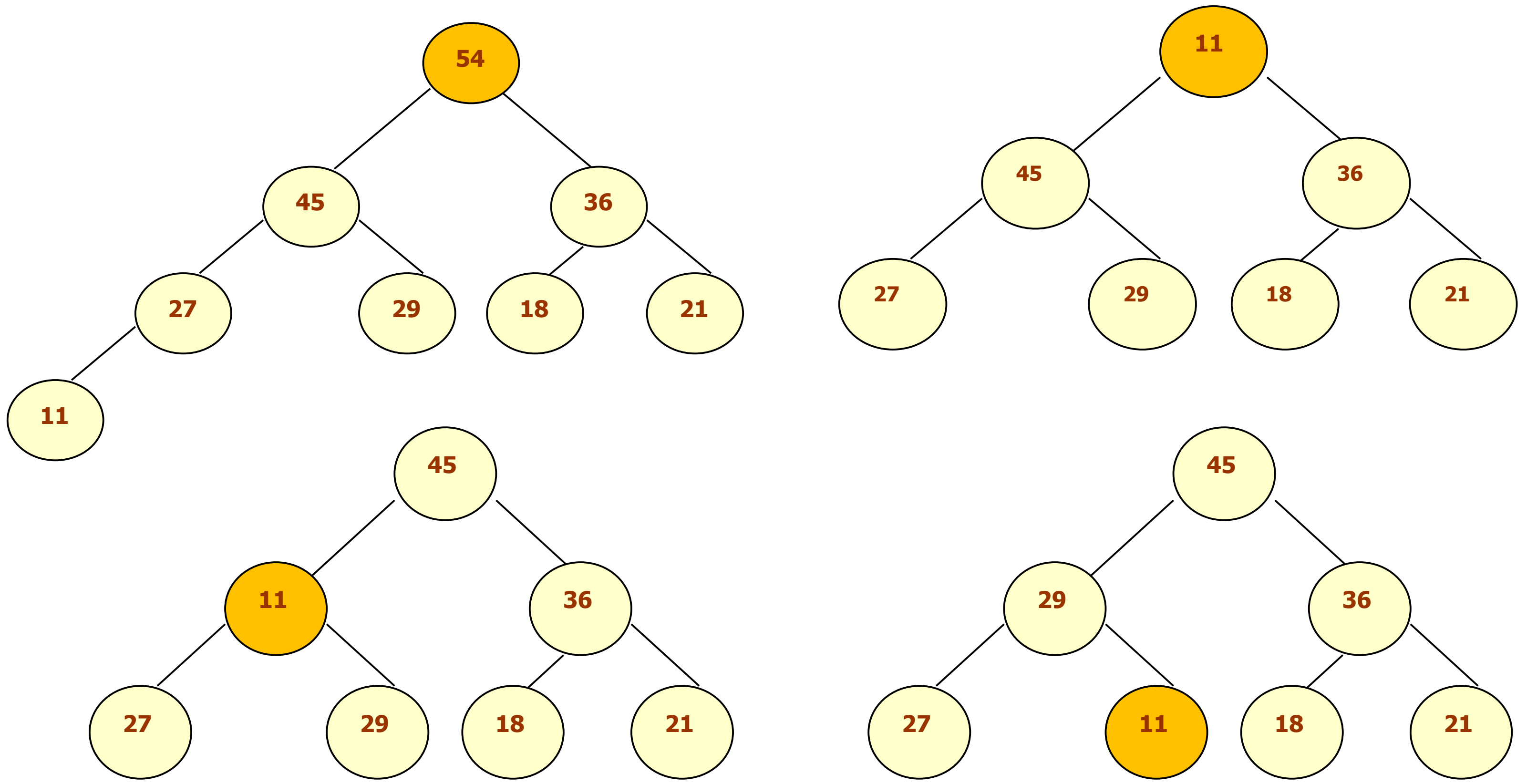
Delete a Node from Heap

Reheapification
Downward

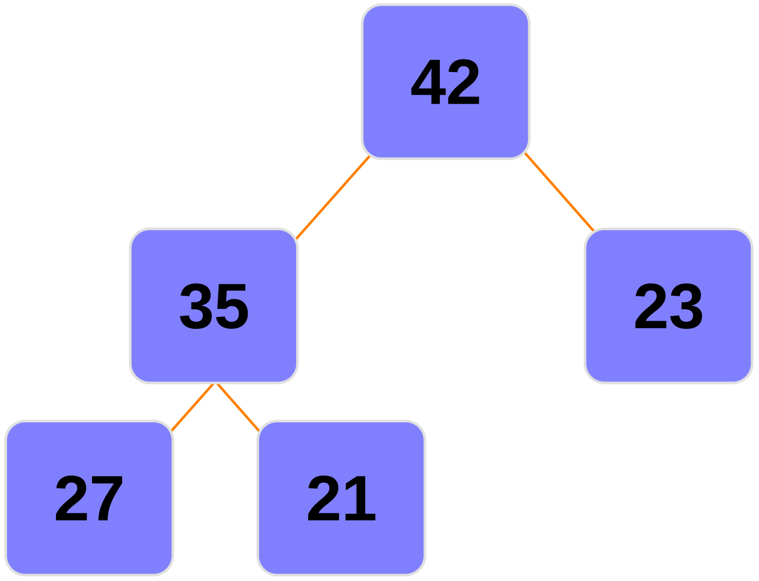


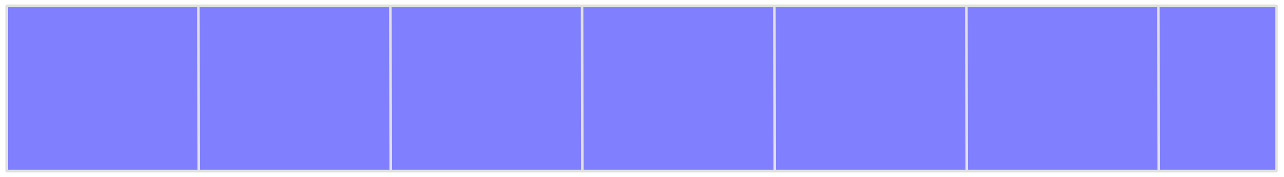
Delete a Node from Heap

Consider the heap H given below and delete the root node's value.



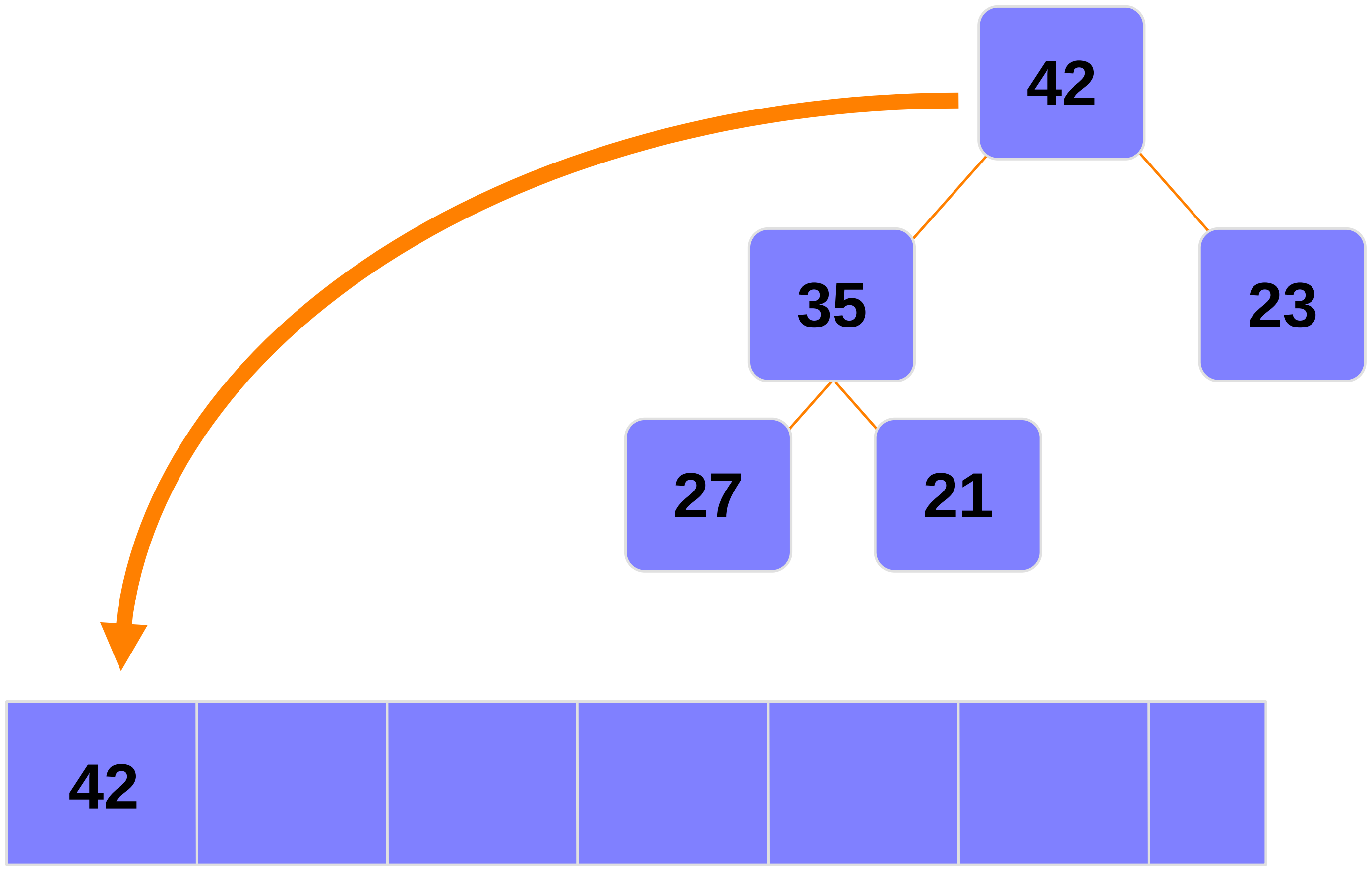
Implementing and Storing a Heap





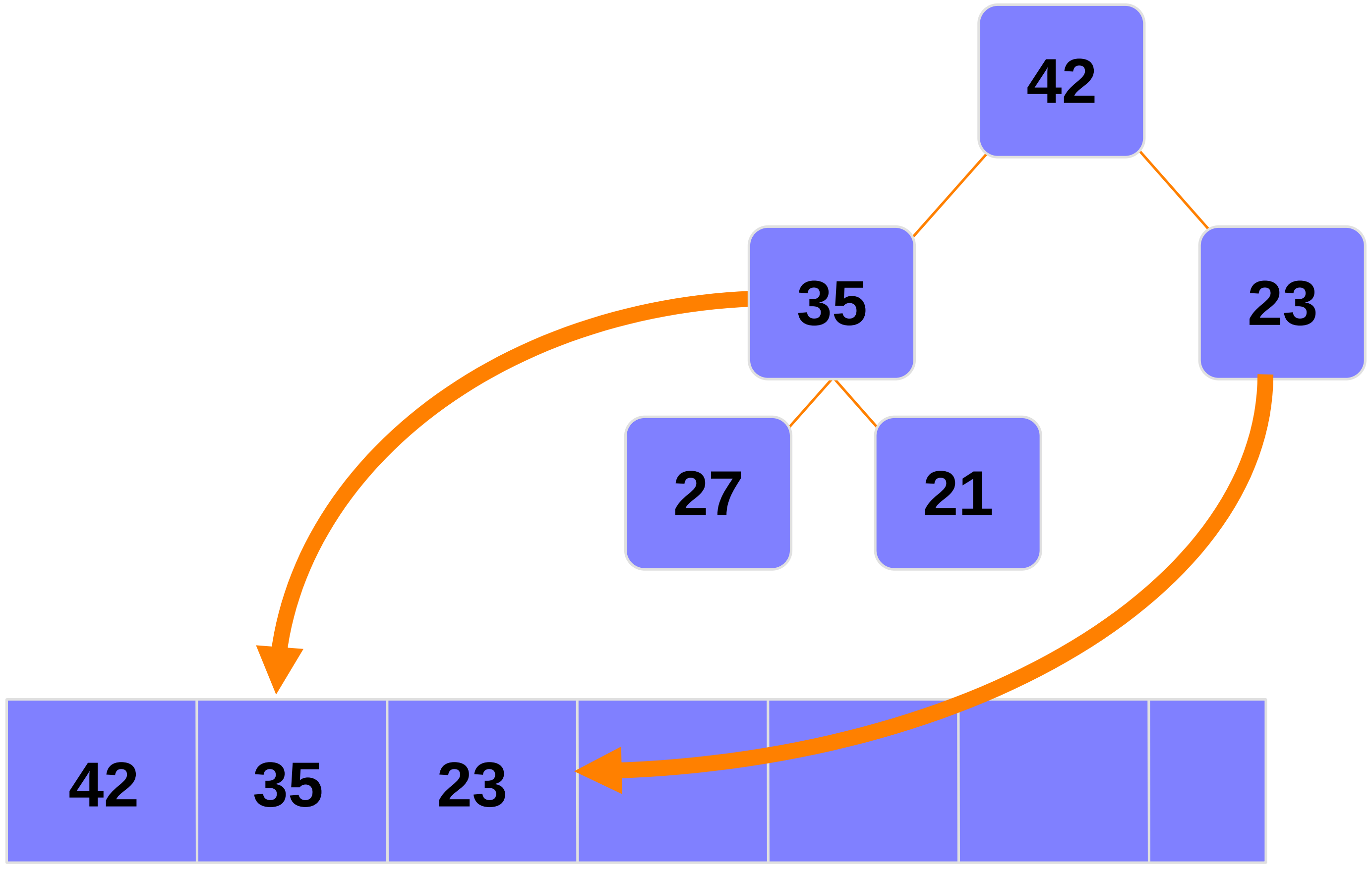
An array of data

Implementing and Storing a Heap



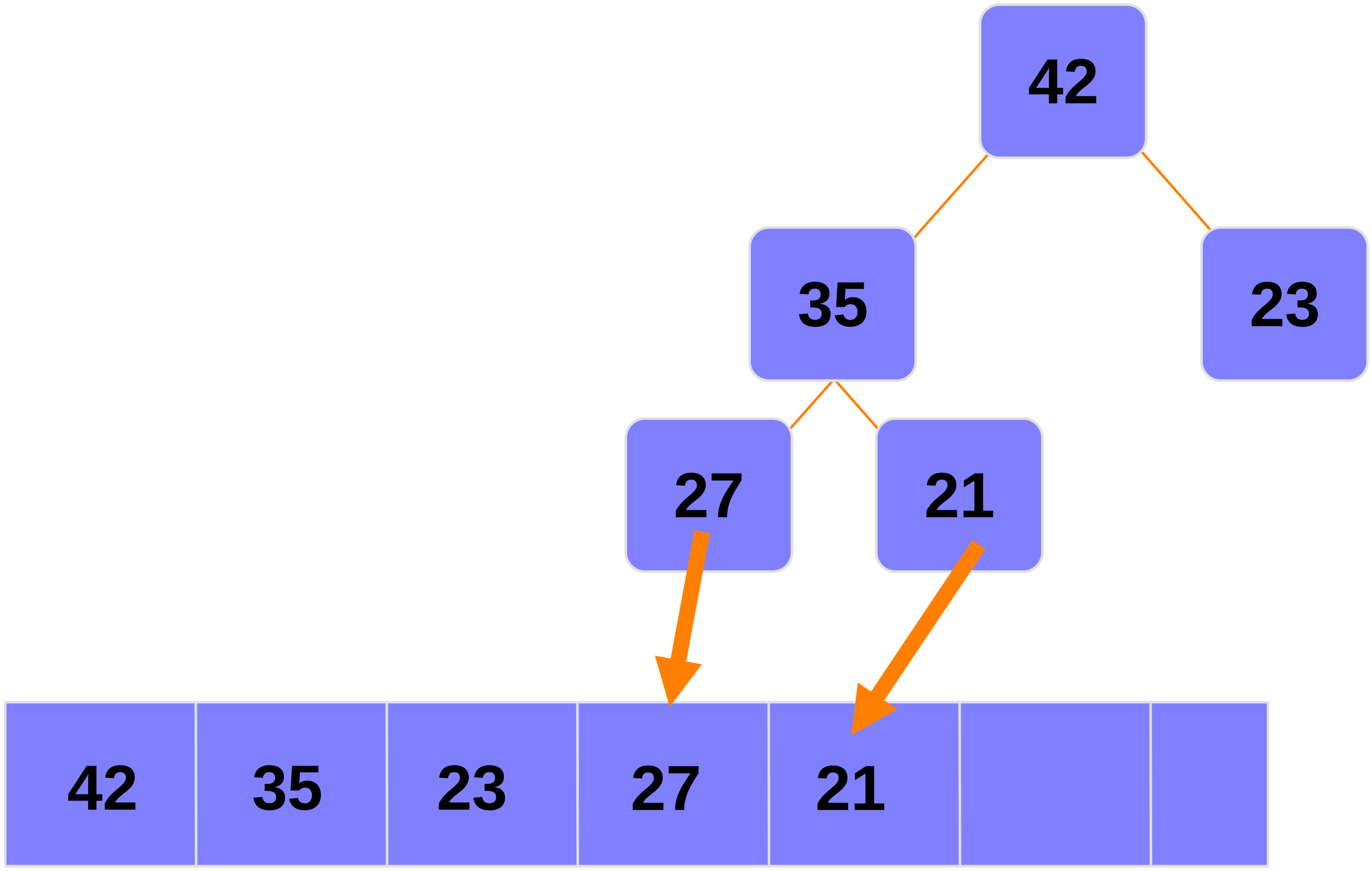
An array of data

Implementing and Storing a Heap



An array of data

Implementing and Storing a Heap



An array of data

Thank You!