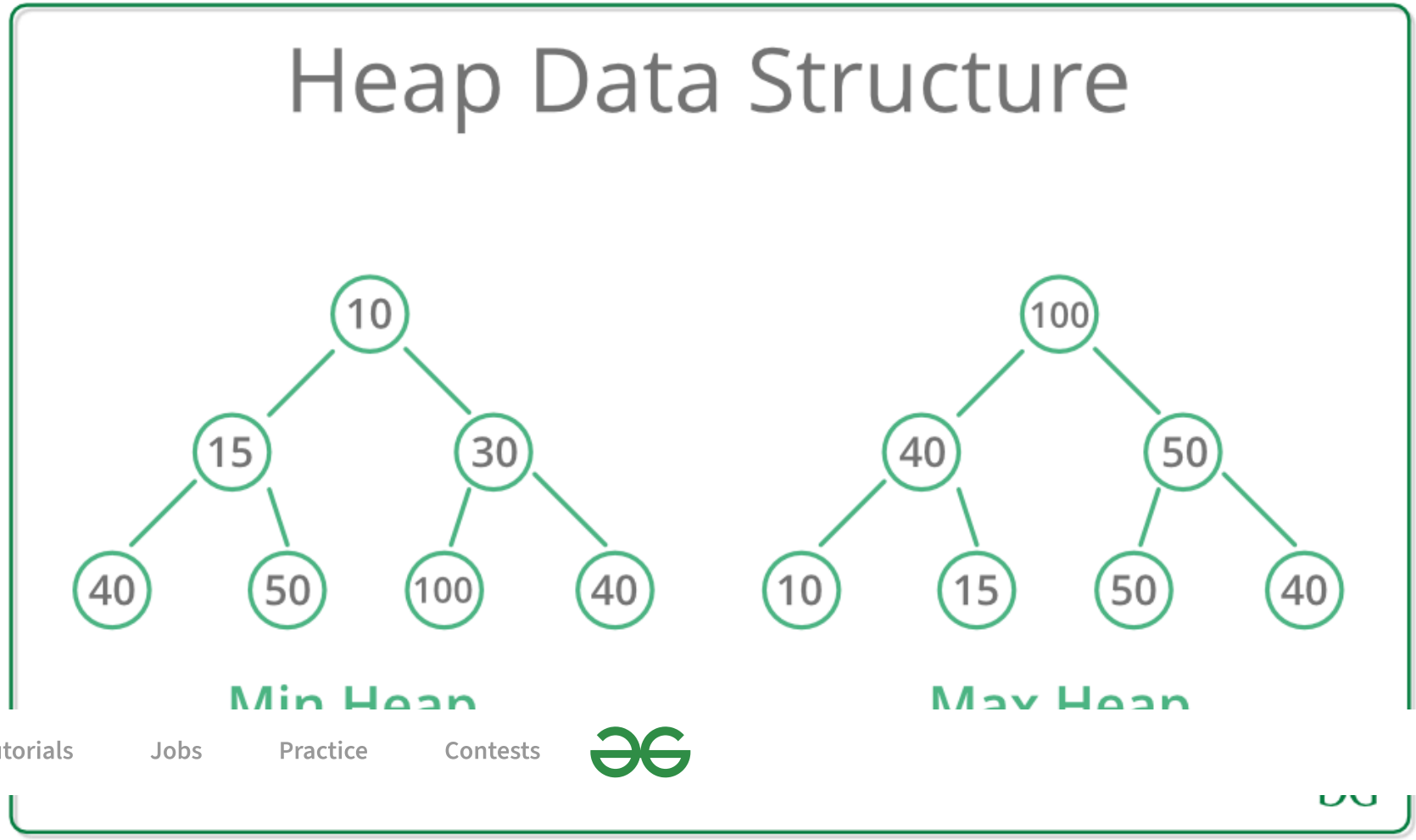
Dash
All
Articles
Videos
Problems
Quiz
Prev
Next

Binary Heap Introduction

A Heap is a Tree-based data structure, which satisfies the below properties:

1. A Heap is a complete tree (All levels are completely filled except possibly the last level and the last level has all keys as left as possible).
2. A Heap is either Min Heap or Max Heap. In a Min-Heap, the key at root must be minimum among all keys present in the Binary Heap. The same property must be recursively true for all nodes in the Tree. Max Heap is similar to MinHeap.

Binary Heap: A Binary Heap is a heap where each node can have at most two children. In other words, a Binary Heap is a complete Binary Tree satisfying the above-mentioned properties.



Representing Binary Heaps

Since a Binary Heap is a complete Binary Tree, it can be easily represented using Arrays.

- The root element will be at Arr[0].
- Below table shows indexes of other nodes for the i^{th} node, i.e., Arr[i]:

Arr[(i-1)/2]	Returns the parent node
Arr[(2*i)+1]	Returns the left child node
Arr[(2*i)+2]	Returns the right child node

Getting Maximum Element: In a Max-Heap, the maximum element is always present at the root node which is the first element in the array used to represent the Heap. So, the maximum element from a max heap can be simply obtained by returning the root node as Arr[0] in O(1) time complexity.

Getting Minimum Element: In a Min-Heap, the minimum element is always present at the root node which is the first element in the array used to represent the Heap. So, the minimum element from a minheap can be simply obtained by returning the root node as Arr[0] in O(1) time complexity.