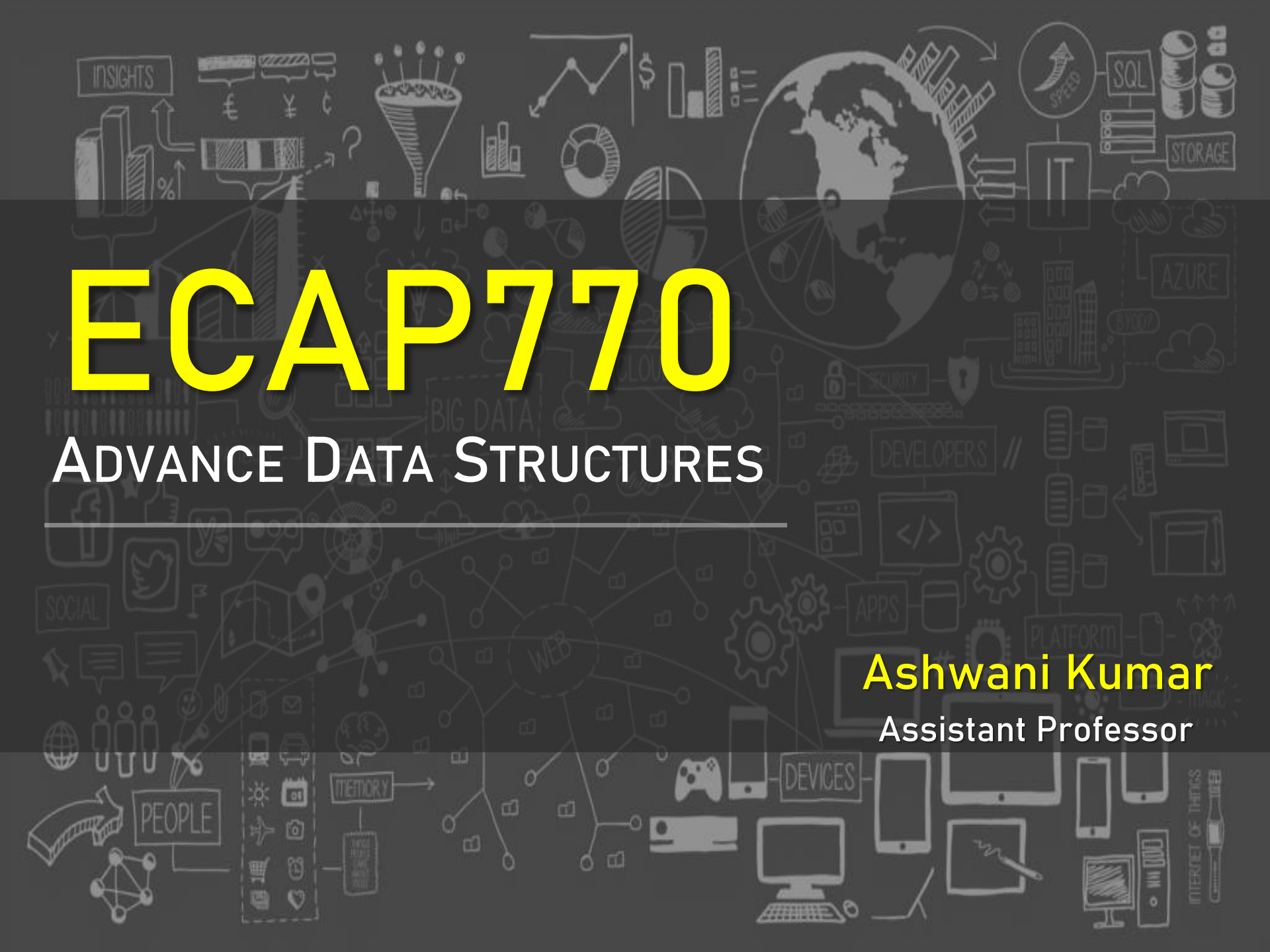


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand Priority queue

Priority queue

- A priority queue is an abstract data type in which each element is associated with a priority value.
- Elements are served on the basis of their priority.
- An element with high priority is dequeued before an element with low priority.
- If two elements have the same priority, they are served according to their order in the queue.

Priority queue

- The priority queue moves the highest priority elements at the beginning of the priority queue and the lowest priority elements at the back of the priority queue.
- It supports only those elements that are comparable.
- Priority queue in the data structure arranges the elements in either ascending or descending order.

Types of Priority Queue

- Ascending Order Priority Queue
- Descending Order Priority Queue

Ascending Order Priority Queue

A ascending order priority queue gives the highest priority to the lower number in that queue

Example:

List: 5 6 20 22 10

Arrange these numbers in ascending order.

List 5 6 10 20 22

Descending Order Priority Queue

A descending order priority queue gives the highest priority to the highest number in that queue.

- Example:
- List: 5 6 35 22 10
- Arrange these numbers
- List: 35 22 10 6 5

Priority Queue Operations

- Inserting an Element into the Priority Queue
- Deleting an Element from the Priority Queue
- Peeking from the Priority Queue (Find max/min)
- Extract-Max/Min from the Priority Queue

Implementation of Priority Queue

- Priority queue can be implemented using
- Array
- Linked list
- Heap data structure
- Binary search tree
- Heap data structure provides an efficient implementation of priority queues.

Complexity analysis of priority queue

Operations	peek	insert	delete
Linked List	O(1)	O(n)	O(1)
Binary Heap	O(1)	O(log n)	O(log n)
Binary Search Tree	O(1)	O(log n)	O(log n)

Binary Heap

- A binary heap tree organises all the parent and child nodes of the tree in a particular order.
- A parent node can have a maximum of 2 child nodes. The value of the parent node could either be:
  - equal to or less than the value of a child node.
  - equal to or more than the value of a child node.

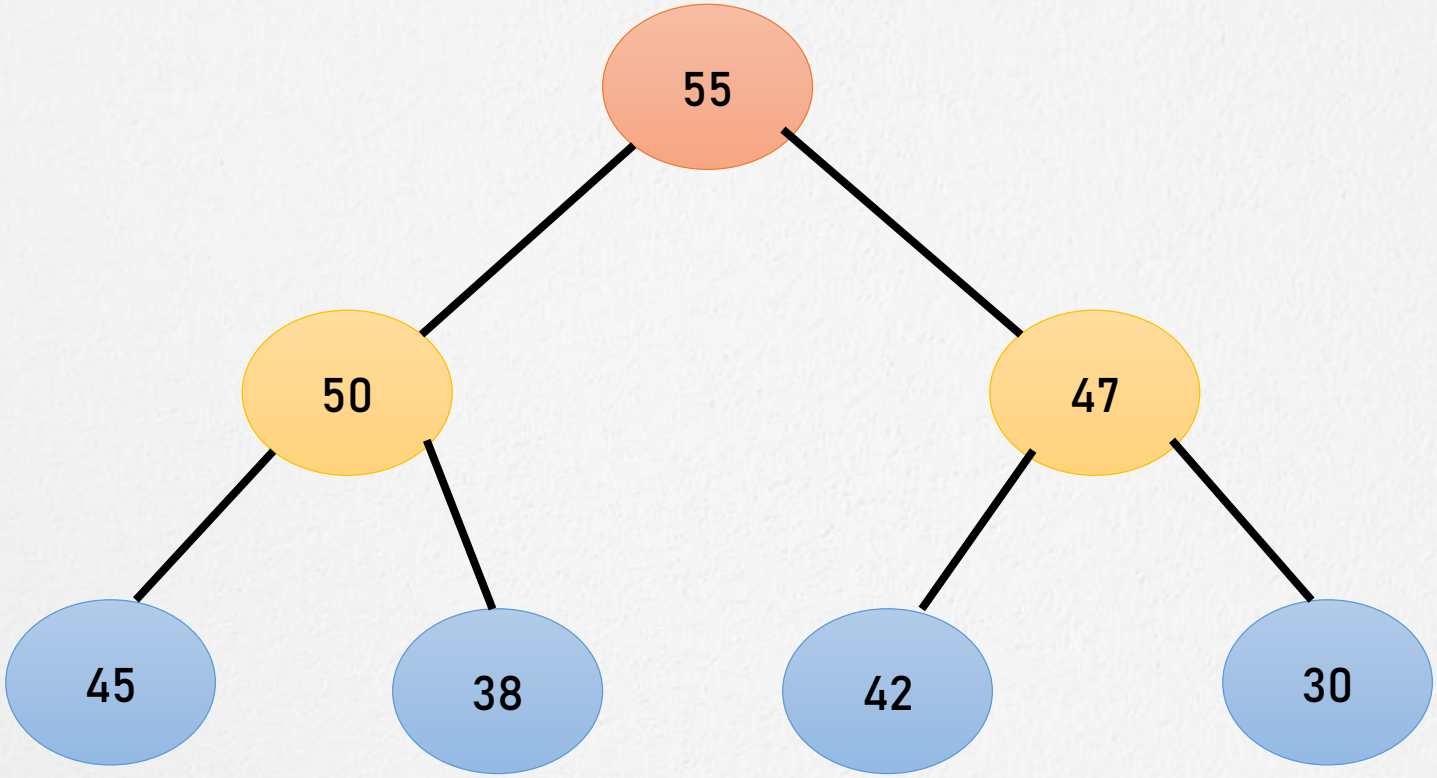
Binary Heap

- Binary heap tree can be divided into two types:
- Max heap
- Min-heap.

Max heap

- The max heap is a binary heap in which a parent node has a value either equal to or greater than the child node value.
- The root node of the tree has the highest value.

Max heap



Algorithm: Insert an Element in a Max Heap

If the tree is empty and contains no node,

    create a new parent node newElement.

else (a parent node is already available)

    insert the newElement at the end of the tree (i.e., last node of the tree from left to right.)

max-heapify the tree

Algorithm: Delete an Element in a Max Heap

If the elementUpForDeletion is the lastNode,

delete the elementUpForDeletion

else replace elementUpForDeletion with the lastNode

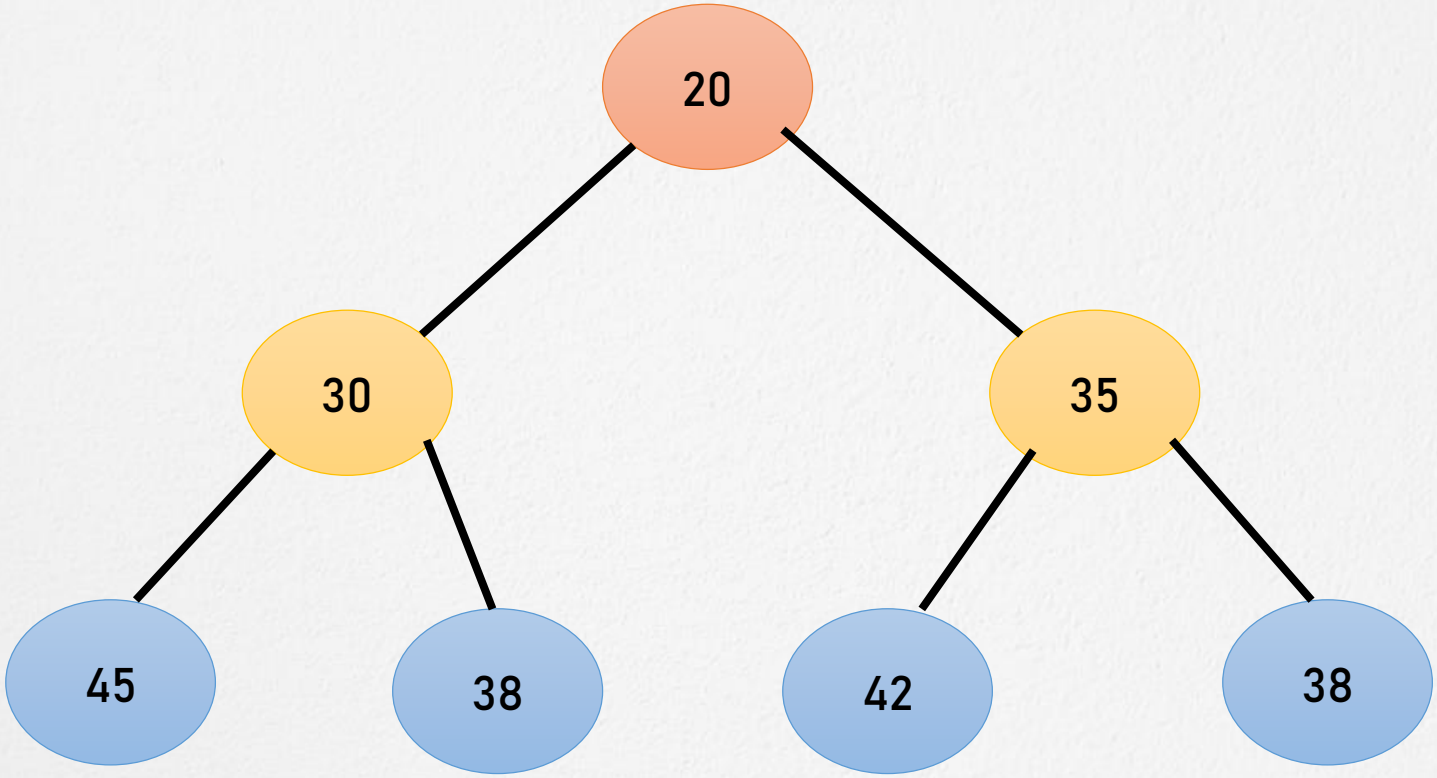
delete the elementUpForDeletion

max-heapify the tree

Min heap

- The min heap is a binary heap in which a parent node has a value equal to or lesser than the child node value.
- The root node of the tree has the lowest value.

Min heap



Priority Queue Applications

- Dijkstra's algorithm: To find shortest path in graph.
- Prim's Algorithm: Prim's algorithm uses the priority queue to the values or keys of nodes and draws out the minimum of these values at every step.

Priority Queue Applications

- Data Compression: Huffman codes use the priority queue to compress the data.
- Operating Systems: load balancing and interrupt handling in an operating system

That's all for now...