بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

In the name of Allah, Most Gracious, Most Merciful

CSE 4303
Data Structure

Topic:Priority Queue, Heap

Asaduzzaman Herok
Lecturer | CSE | IUT
asaduzzaman34@iut-dhaka.edu

Priority Queue

With queues

- The order may be summarized by first in, first out

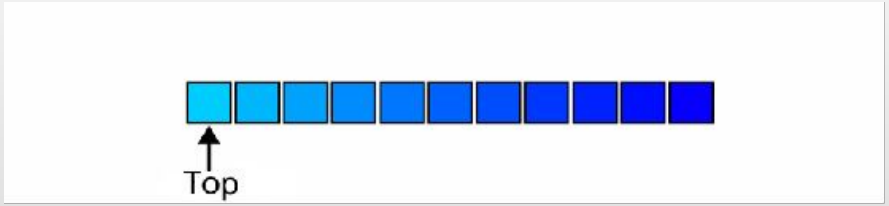
But, if each object is associated with a priority, we may wish to pop that object which has highest priority

- With each pushed object, we will associate a nonnegative integer (0, 1, 2, ...) where:
- The value 0 has the highest priority, and
- The higher the number, the lower the priority
- Other way 0 can have the lowest priority and higher number has higher priority.

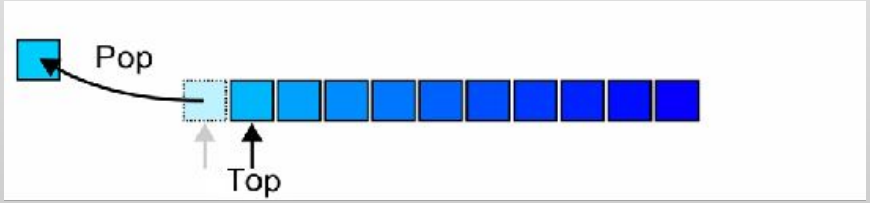
A Priority Queue is a particular type of data structure in which every element is assigned a priority. Elements with higher priorities are dequeued before those with lower ones. In the event of similar priorities, elements are dequeued based on their existing order in the queue.

Operations

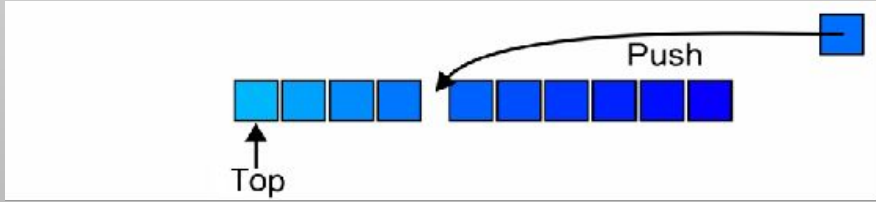
- The top of a priority queue is the object with highest priority



- Popping from a priority queue removes the current highest priority object:



- Push places a new object into the appropriate place



Lexicographical Priority

Priority may also depend on multiple variables:

- Two values can specify a priority: (a, b)
  - A pair (a, b) has higher priority than (c, d) if:
    - a < c, or
    - a = c and b < d

- For example, (5, 19), (13, 1), (13, 24), and (15, 0) all have higher priority than (15, 7)

Priority Queue Applications

- Any event/job management that assign priority to events/jobs
- Priority-based OS process scheduler
- Event-driven simulation (traffic flows)
- Artificial intelligence search algorithms
- Used in the Dijkstra's shortest path algorithm.
- Used in prim's (minimum spanning tree) algorithm
- Used in data compression techniques like Huffman code.

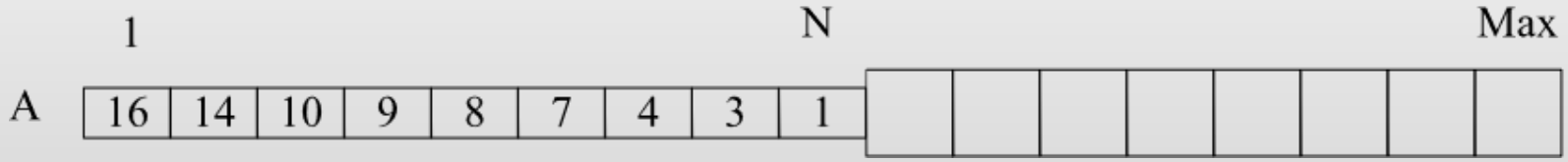
Implementations of Priority Queues

- Array Implementation
- Linked List Implementation
- Heap Implementation
  - Array based
  - Linked List based

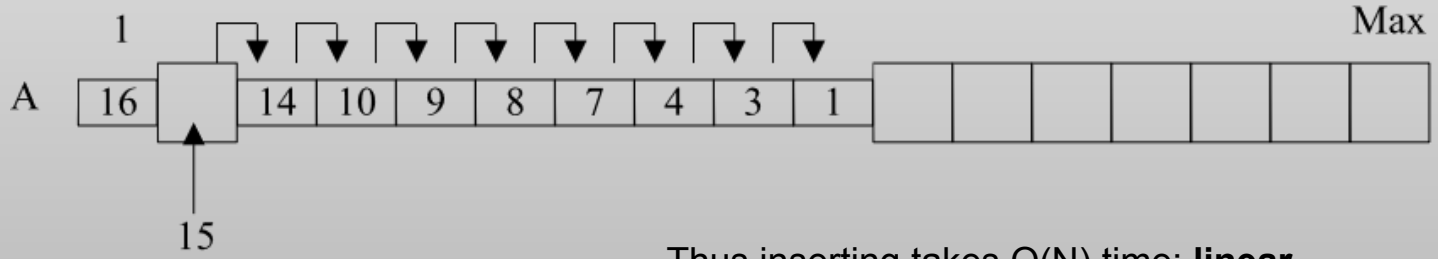
Array Implementation of Priority Queues

Suppose items with priorities 16, 14, 10, 9, 8, 7, 4, 3, 1 are to be stored in a priority queue.

One implementation:

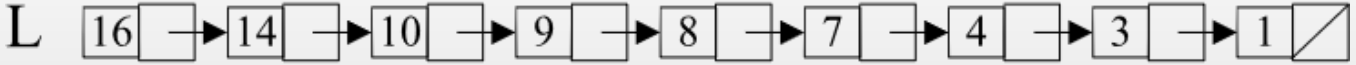


Suppose an item with priority 15 is added:

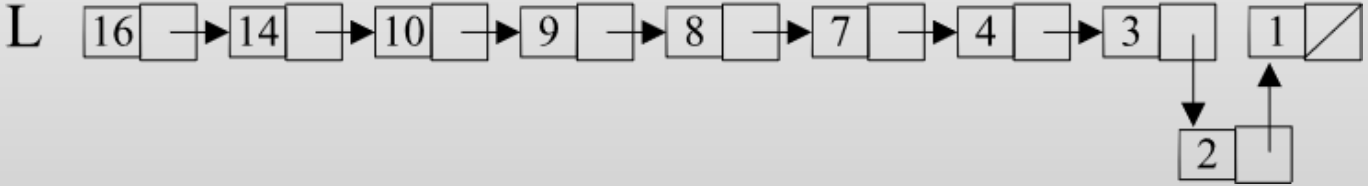


Thus inserting takes O(N) time: **linear**

Linked List Implementation of Priority Queues



Suppose an item with priority 2 is to be added:



Only O(1) (constant) pointer changes required, but it takes O(N) pointer traversals to find the location for insertion.

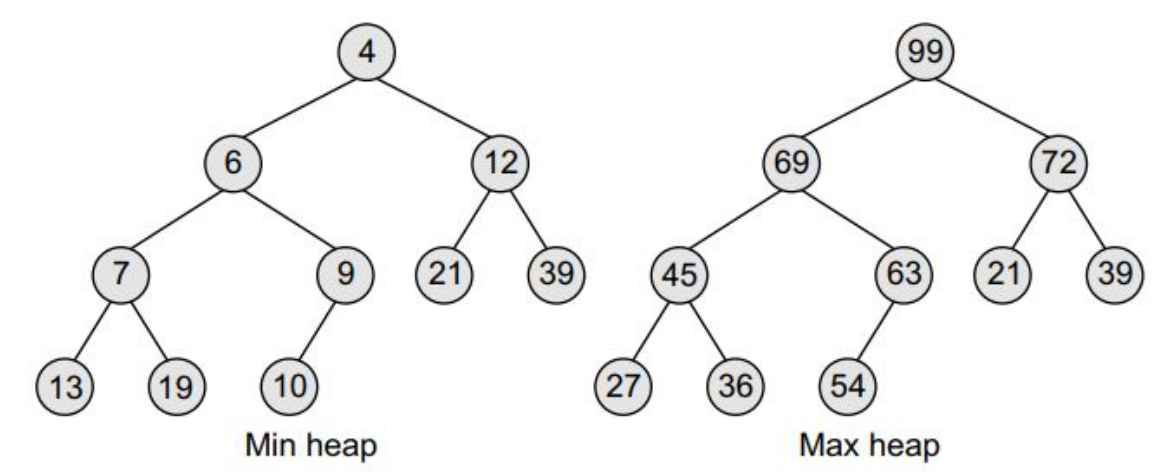
Wanted: a data structure for PQs that can be both **searched** and **updated** in better than O(N) time.

Binary Heaps

A binary heap is a complete binary tree in which every node satisfies the heap property which stats that, for

- Max Heap: If B is a child of A, then key(A) ≥ key(B)
- Min Heap: If B is a child of A, then key(A) ≤ key(B)

This implies that elements at every node will be either greater(max heap) / less(min heap) than or equal to the element at its left and right child.



Implementation of Binary Heaps

Linked List Based:

```
struct Node
{
    Node *parent;
    int val;
    Node *left;
    Node *right;
};
```

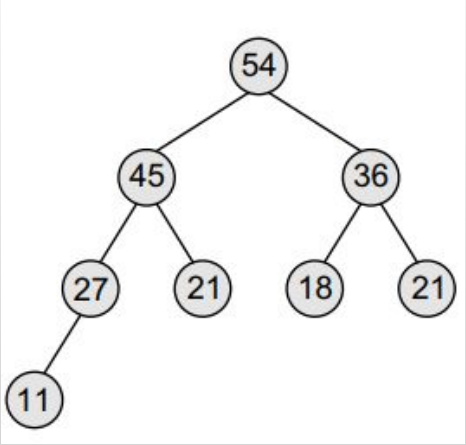
Array Based: index starting from 1.

Root: arr[1]

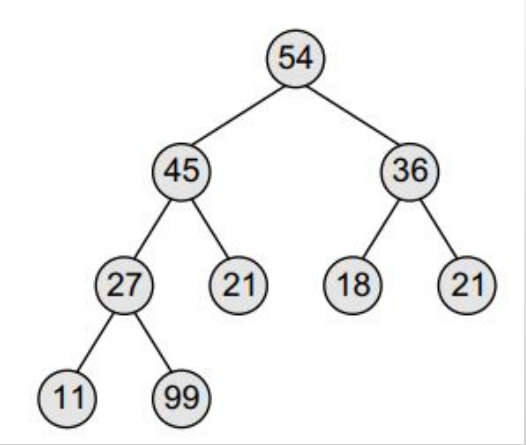
Node: value at i =	arr[i]
Parent:	arr[i/2]
Left child:	arr[2*i]
Right child:	arr[2*i+1]

Pseudomonas implementation details will be shown in board

Inserting a New Element in a Binary Heap

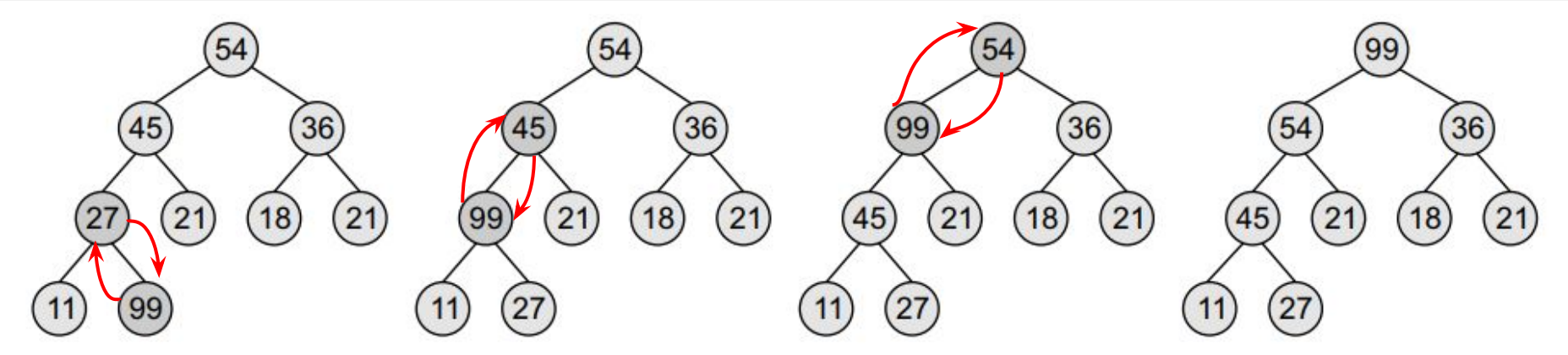


Existing Heap
(max heap)



Insert new node(99) at a leaf node

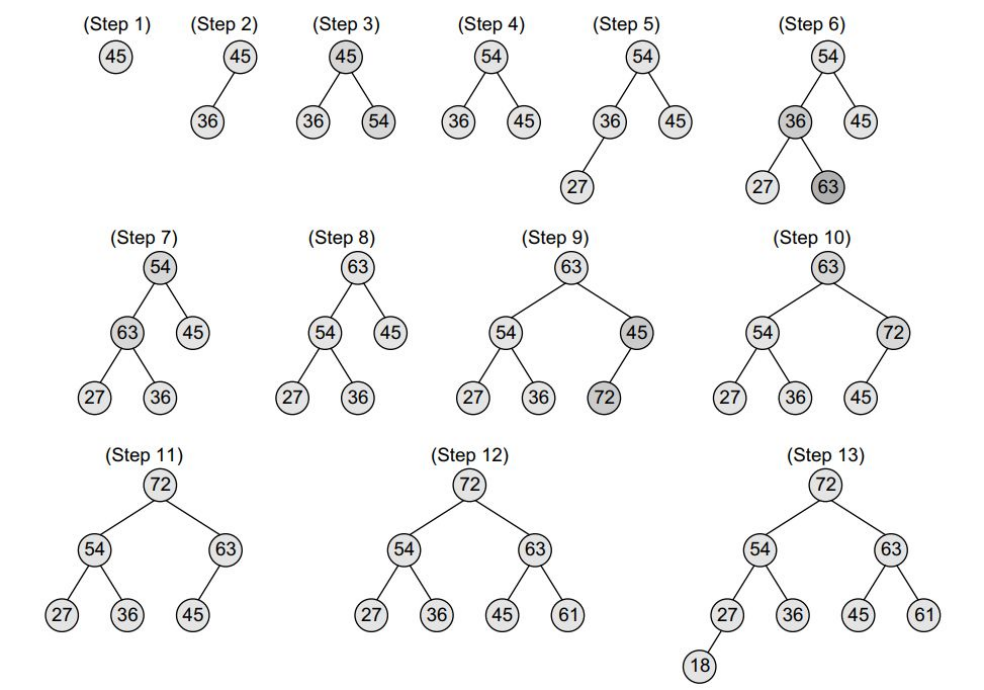
Inserting a New Element in a Binary Heap



Heapify the binary heap (max heap)

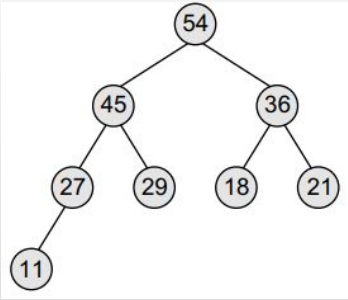
Creating a max heap

Build a max heap H from the given set of numbers: 45, 36, 54, 27, 63, 72, 61, and 18. Also draw the memory representation of the heap.



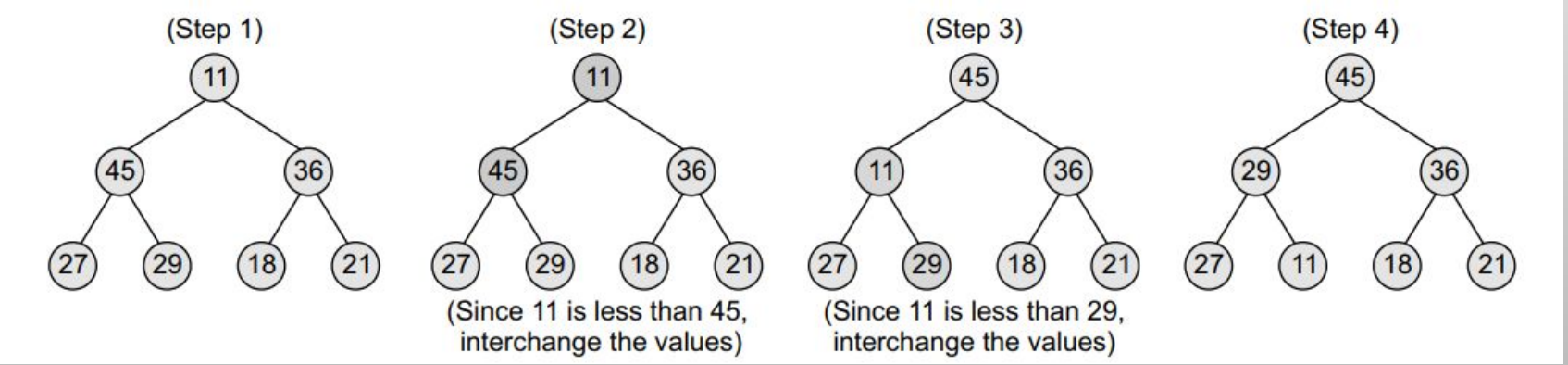
Deleting an Element from a Binary Heap

1. Replace the root node's value with the last node's value so that heap is still a complete binary tree but not necessarily a heap.
2. Delete the last node.
3. Sink down the new root node's value so that H satisfies the heap property. In this step, interchange the root node's value with its child node's value (whichever is largest among its children).



Given Heap (max heap)

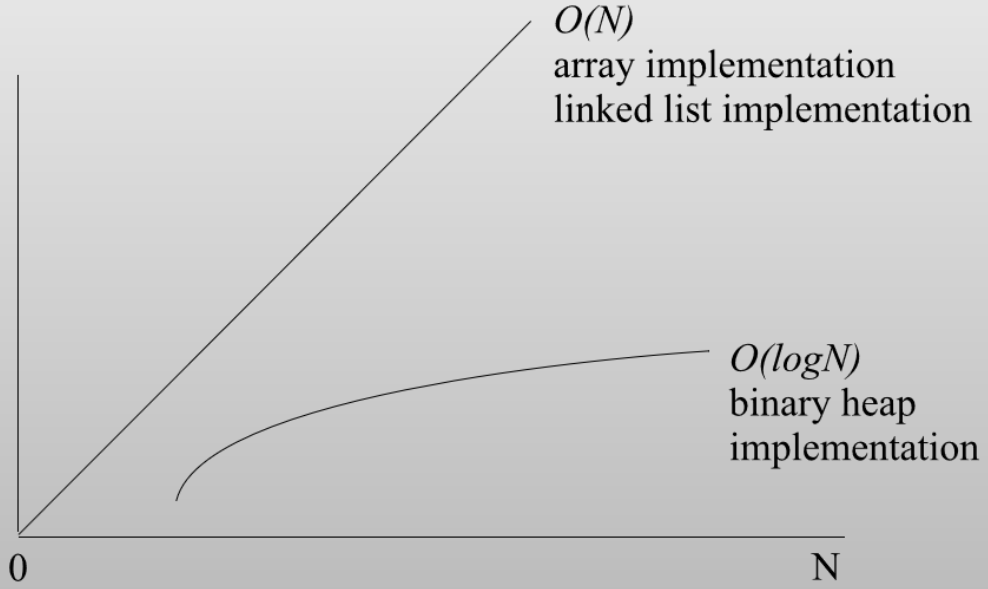
Deleting the root



Complexity of Binary Heap Insertion & Deletion

What is the maximum number of parent shifts necessary?

The height of the binary heap = floor(log2N)
So binary heap add can be done in O(logN) time.



Acknowledgements

Rafsanjany Kushol
PhD Student, Dept. of Computing Science,
University of Alberta

Sabbir Ahmed
Assistant Professor
Department of CSE, IUT