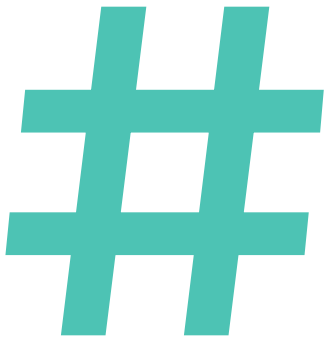

Priority Queue

Step 01 Priority Queue

A priority queue is a data structure that stores an element based on its value priority. Either the highest or lowest priority element is dequeued/processed first.

In emergency rooms, patients wait in line or queue, and they are treated based on the highest priority, and not the order in which they arrive. Well, we have a data structure called priority queue which behaves the same as an emergency queue, the element with the highest/lowest value gets processed first.

> NOTE
> Priority Queue elements are sorted based on the highest/lowest value.

Step 02 Implementations

There are several ways to implement a priority queue, and the most common ways are:

01 Binary Heap(Sorted Heap)

02 Sorted Linked List

03 Balanced Binary Search Tree

Step 03 Implement Priority Queue using Sorted Linked List

In the following implementation, the lowest value will be dequeued first.

```
// Node class to represent each element in the priority queue
 public class Node {
    int priority; // priority value
    int value; // actual value
    Node next;
    Node(int priority, int value) {
        this.priority = priority;
        this.value = value;
        this.next = null;
    }
}

// The priority queue class
public class PriorityQueue {
    private Node head = null;
    public void insert(int priority, int value) {
        Node newNode = new Node(priority, value);
        if(head == null || head.priority > priority) {
            newNode.next = head;
            head = newNode;
        } else {
            Node current = head;
            while (current.next != null && current.next.priority < priority) {
                current = current.next;
            }
            newNode.next = current.next;
            current.next = newNode;
        }
    }
    public int delete() {
        if (head != null) {
            int value = head.value;
            head = head.next;
            return value;
        } else {
            throw new RuntimeException("Queue is empty");
        }
    }
    public static void main(String[] args) {
        PriorityQueue pq = new PriorityQueue();
        pq.insert(3, 30);
        pq.insert(1, 10);
        pq.insert(2, 20);
        System.out.println(pq.delete()); // 10
        System.out.println(pq.delete()); // 20
        System.out.println(pq.delete()); // 30
    }
}
```

OUTPUT

```
10
20
30
```