E.

Date:- / /
Page No.:- 31

Title:- priority Queue Implementation for Efficient patient service management in a Hospital.

problem statement:- consider a scenario for Hospital to cater services to diffrent kinds of patients as serious (top priority) b] non-serious (medium priority) c] General checkup. (least priority) Implement the priority queue to clear ~~out~~ cater services to the patients.

Software Requirement:- g++ / gcc compiler 64 bit fedora, eclipse IDE

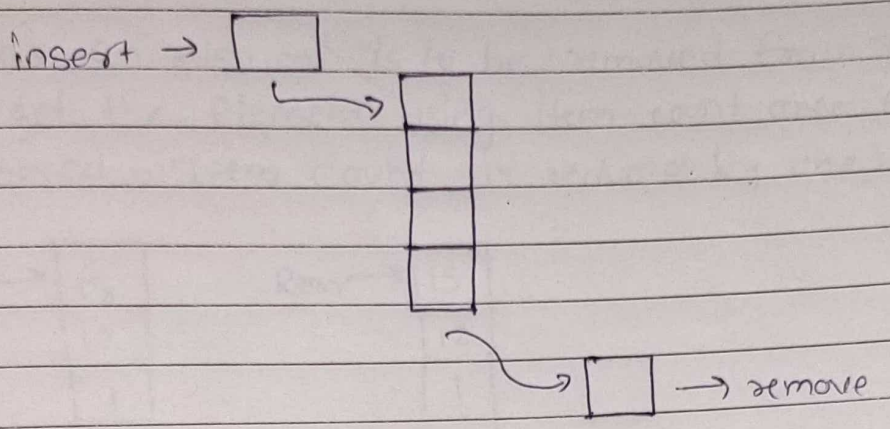
Theory:- A priority queue can be used to efficiently cater to different kinds of patients based on their priority levels. A priority queue is a data structure that stores elements along with their associated priorities.

priority Queue is more specialized data structure than Queue like ordinary queue, priority queue has same method but with a major difference In priority queue items are ordered by key value so that item with the highest value of key is at rear or vice versa. So we are assigned priority to item based on its key value. lower the value, higher the priority following are the principal methods of a priority Queue

1] insert / enqueue :- add an item to the rear of the queue.

2] remove / dequeue - remove an item from the front of the queue.

- priority Queue presentation

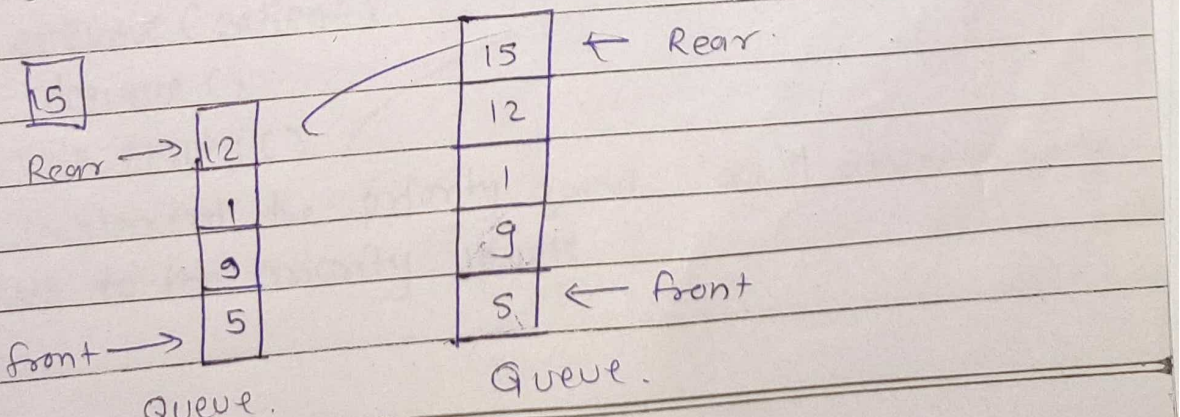


- we are going to implement Queue using array in this article. There is few more operations supported by queue which are following

1. peek - get the element at front of the queue.
2. isfull - check if queue is full
3. isEmpty - check if queue is empty.

• Insert / Enqueue operation

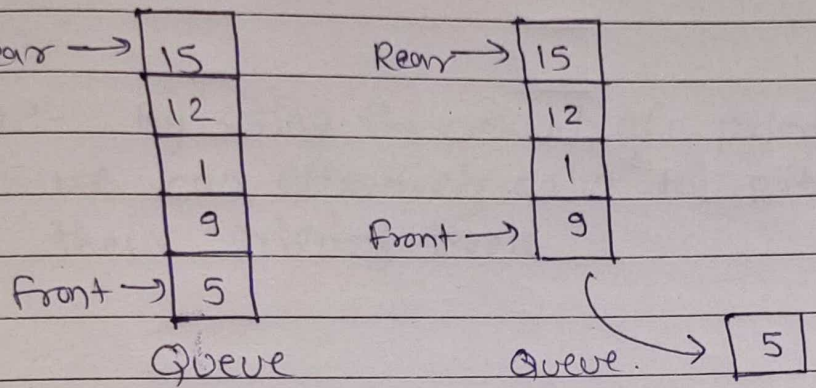
- whenever an element is inserted into queue, priority queue inserts the items according to its order Here we're assuming that data with value has low priority



One item inserted at rear

- Remove / Dequeue operations.

- whenever an element is to be removed from queue queue get the element using item count once element is removed. Item count is reduced by one.



one item removed from front.

Algorithm :-

1. Start
2. Create a class or data structure to represent a patient with the attributes like name, priority of patient.
3. create a priority queue data structure to store patients information.
4. Define the following operations for the priority queue
   a. enqueue (patient)
   b. dequeue ()
   c. is empty ()
5. To implement the priority queue, we'll assign numerical values to the priority levels.
6. stop.

- The time complexity of enqueueing a patient into the priority queue is O(log n) and for dequeueing is O(log n) as well
- The space complexity of the priority queue is O(n), where n is number of patients in the queue.

Conclusion :- By using the concept of priority queue algorithm, we can effectively cater to patients based on their priority levels.