Modern Education Society's Wadia College of Engineering, Pune

210256: DATA STRUCTURES and ALGORITHM LABORATORY (2019 C0URSE)

NAME OF STUDENT:	**CLASS:**
SEMESTER/YEAR:	**ROLL NO:**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED BY:	**EXPERIMENT NO: E20**

TITLE: PRIORITY QUEUE

AIM/PROBLEM STATEMENT: Consider a scenario for hospital to cater services to different kind of patients as Serious (top priority), non-serious (Medium Priority), and General checkup (Least Priority). Implement priority queue to cater services to the patients.

OBJECTIVES:

1. To understand priority queue data structure.
2. To understand practical implementation and usage of queue linear data structures

OUTCOMES:

1. Apply and analyze appropriate data structure to solve the real time problems using priority queue

PRE-REQUISITE:

1. Knowledge of C++ programming
2. Knowledge of 2D-Array, structures

THEORY:

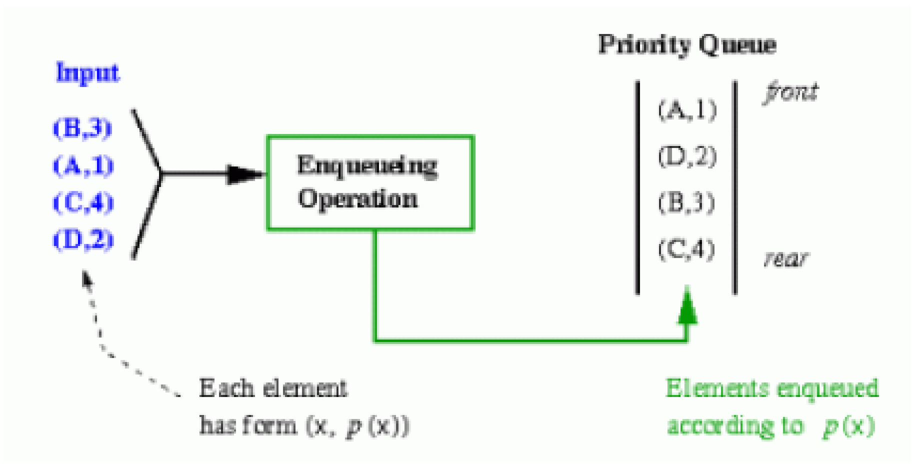
A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

A queue in which we are able to insert and remove items from any position based on some property (such as priority of the task to be processed) is often referred as priority queue. Fig 1 represents a priority queue of jobs waiting to use a computer.

Priority Queue is an extension of queue with following properties.

- Every item has a priority associated with it.
- 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.

In the below priority queue example is given:



A typical priority queue supports following operations.
insert(item, priority): Inserts an item with given priority.
getHighestPriority(): Returns the highest priority item.
deleteHighestPriority(): Removes the highest priority item.

How to implement priority queue?

Using Array: A simple implementation is to use array of following structure. insert() operation can be implemented by adding an item at end of array in O(1) time.
getHighestPriority() operation can be implemented by linearly searching the highest priority item in array. This operation takes O(n) time.

deleteHighestPriority() operation can be implemented by first linearly searching an item, then removing the item by moving all subsequent items one position back.

Using Heaps:

Heap is generally preferred for priority queue implementation because heaps provide better performance compared arrays or linked list.

In a Binary Heap, getHighestPriority() can be implemented in O(1) time, insert() can be implemented in O(Logn) time and deleteHighestPriority() can also be implemented in O(Logn) time.With Fibonacci heap, insert() and getHighestPriority() can be implemented in O(1) amortized time and deleteHighestPriority() can be implemented in O(Logn) amortized time.

Applications of Priority Queue:

1) CPU Scheduling

2) Graph algorithms like Dijkstra’s shortest path algorithm, Prim’s Minimum Spanning Tree, etc

3) All queue applications where priority is involved

QUESTIONS:

1. Define Ascending and Descending Priority queue.
2. Describe various applications of Priority Queue.
3. Write short note on ADT of Priority queue