Assignment No. 12

Problem Statement:

Write program to implement a priority queue in C++ using an inorder List to store the items in the queue. Create a class that includes the data items(which should be template) and the priority (which should be int)The inorder list should contain these objects ,with operator <= overloaded so that the items with highest priority appear at the beginning of the list (which will make it relatively easy to retrieve the highest item.)

Theory

Priority Queue

Priority queue is an abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according to their order in the queue.

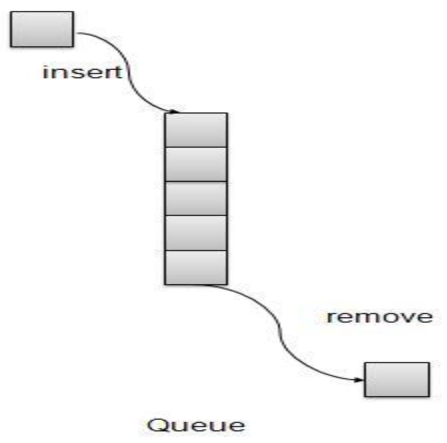
Priority Queue is more specialized data structure than Queue. Like ordinary queue, priority queue has same method but with a major difference. In Priority queue items are ordered by key value so that item with the lowest value of key is at front and item with the highest value of key is at rear or vice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

An important application of a priority queue is sorting
• PriorityQueueSort (collection S of n elements)
• put the elements in S in an initially empty priority queue by means of a series of n insert() operations on the pqueue, one for each element
• extract the elements from the pqueue by means of a series of n removeMin() operations

Basic Operations

- insert / enqueue − add an item to the rear of the queue.
- remove / dequeue − remove an item from the front of the queue.

Priority Queue Representation



There are few more operations supported by queue which are following.

- getHighestPriority(): Returns the highest priority item.
- Peek − get the element at front of the queue.
- isFull − check if queue is full.
- isEmpty − check if queue is empty.

Using Array: We're going to implement Queue using array. A simple implementation is to use array of following structure.

```
Struct item{

Int item;

Int priority;

}
```

insert() operation can be implemented by adding an item at end of array in O(1) time.

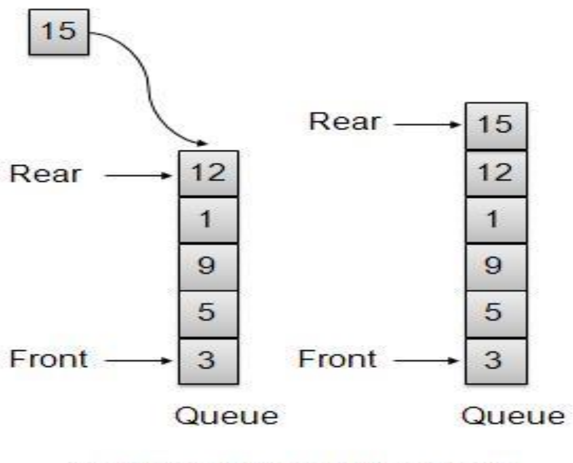
getHighestPriority() operation can be implemented by linearly searching the highest priority item in array. This operation takes O(n) time.

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

We can also use Linked List, time complexity of all operations with linked list remains same as array. The advantage with linked list is deleteHighestPriority() can be more efficient as we don’t have to move items.

Insert / Enqueue Operation

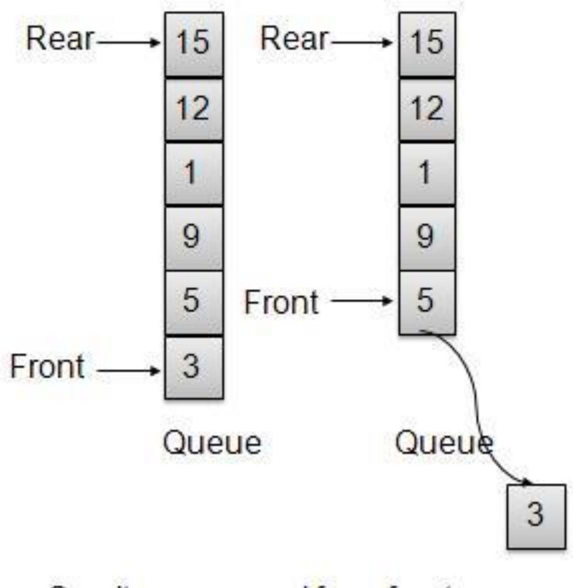
Whenever an element is inserted into queue, priority queue inserts the item according to its order. Here we're assuming that data with high value has low priority.



One item inserted at rear end

Remove / Dequeue Operation

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

Conclusion

Hence, we learnt to implement a priority queue in C++ using an inorder.