Book

A Simplified Approach

to

Data Structures

Prof.(Dr.)Vishal Goyal, Professor, Punjabi University Patiala

Dr. Lalit Goyal, Associate Professor, DAV College, Jalandhar

Mr. Pawan Kumar, Assistant Professor, DAV College, Bhatinda

Shroff Publications and Distributors

Edition 2014

<<Some Special Kinds of Queues>>

Contents

- Some special kinds of queues
- Deque
- Priority Queue
- Double Ended Priority Queue
- Applications of Priority Queues
- Applications of Queues

Some special kinds of queues

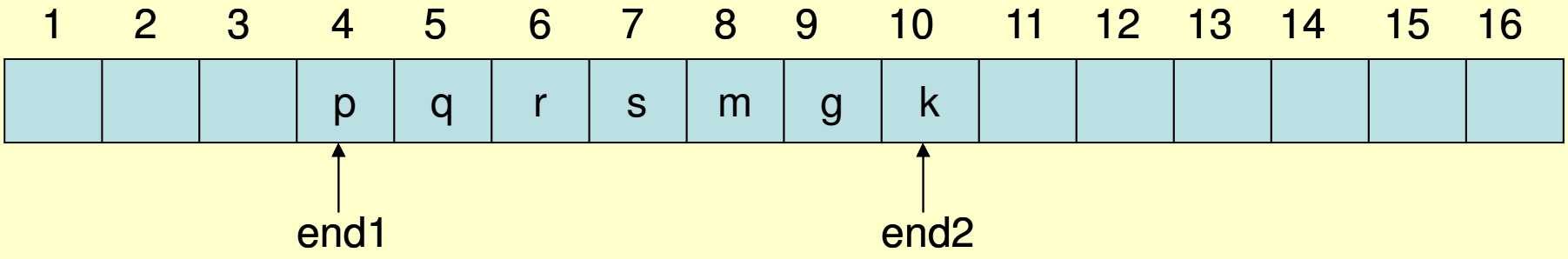
There are some special kinds of queues which have some specialty while performing the insertion and deletion operations. These special kinds of queues are:

- **Deque**
- **Priority Queue**
- **Double Ended Priority Queue**

Deque (Double Ended Queue)

Double ended queue is the linear queue data structure in which insertion and deletion operations are not restricted to one end but rather insertion and deletion operations can be perform on either of the two ends. Deque data structure is also known as Deck.

The representation of deque data structure can be shown diagrammatically as:



Deque with 7 elements

Deque (continued...)

Instead of using the notation of Front and Rear, we use two variables end1 and end2 to represent the index of two ends of the queue. An element can be inserted either at end1 or end2. Similarly, the element can be deleted either at end1 or end2.

In the above shown deque, the insertion of new element can take place at index 3 or at index 11. Similarly, the element present at index 4 or index 10 can be deleted.

Categories of deque

A deque can be categorized into two categories:

- Input Restricted Deque
- Output Restricted Deque

In case of **input restricted deque**, the insertion operation is restricted to one end but the deletion can take place at either end of the list.

In case of **output restricted** deque, the deletion operation is restricted to one end but the insertion can take place at either end of the list.

Priority Queue

Priority queue is the special kind of queue data structure in which insertion and deletion operations are performed according to some special rule rather than just FIFO rule. In case of priority queue, a priority number is associated with each element. The elements are inserted and deleted according to this priority number. The following two rules are applied to process the elements in the priority queue:

1. The elements with higher priority are processed before the elements with lower priority.

2. In case of elements with same priority, elements are processed according to the First In First Out (FIFO) rule.

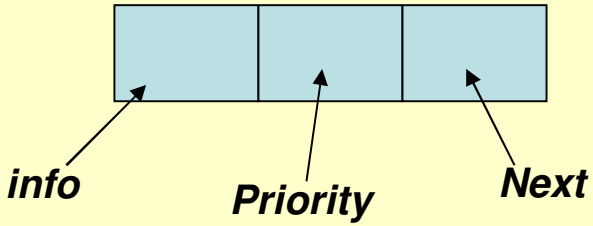
Priority Queue (continued...)

Priority queues can be represented into the memory in following ways:

- Priority Queue using linked list
- Priority Queue using multiple queues
- Priority Queue using heap structure

Priority Queue using linked list

In linked list representation of priority queue, each node of the linked list is divided into three parts as shown below:



Info part holds the element of the queue.
Priority part holds the priority number of the element.
Next part holds the address of next node of linked list

In linked list representation of priority queue, insertion of an element takes place according to the priority number of the element where as the deletion takes place from front end of linked list.

Priority Queue using multiple queues

In this representation of priority queue, a separate queue is maintained for all the element with same priority level which follows the general FIFO order. These separate queues may be circular queues for its efficient use and will also have separate variables ***Front*** and ***Rear***.

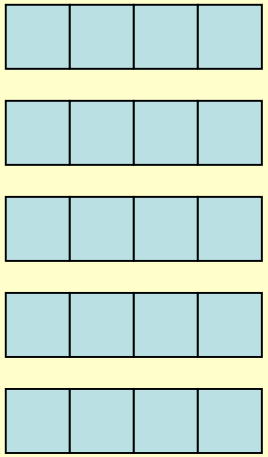
✓ Following information must be known in advance while using multiple queues representation

- The maximum number of priority levels
- The maximum number of elements with same priority

Priority Queue representation using multiple queues

This is representation of Priority Queue using multiple queues

Priority	Front	Rear
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0



See an example on next slide after inserting and deleting elements what will be the resulting representation.

Priority Queue representation using multiple queues

EXAMPLE

Lets take an example, After inserting and deleting following elements what the resulting representation will be:

Insert p with priority 3
Insert n with priority 5
Insert g with priority 3
Insert s with priority 4
Insert m with priority 3
delete an element
Insert k with priority 4
insert d with priority 3
delete an element
Insert b with priority 2
Insert i with priority 2
delete an element
Insert r with priority 1
Insert v with priority 2

Priority	Front	Rear	1	2	3	4
1	1	1	r			
2	2	3	b	i	v	
3	2	4	p	g	m	d
4	1	2	s	k		
5	1	1	n			

Priority Queue representation using heap structure

Both the heap types (Max heap and Min heap) can be used for implementing priority queue. Because the root element in case of ***Max heap*** is always the largest element. This means that if we have largest element as the highest priority element then it is easy to process because we know that it is at the root node of the tree.

Similarly, if the smallest element is our highest priority element then we can process it using ***Min heap***.

Double Ended Priority Queue

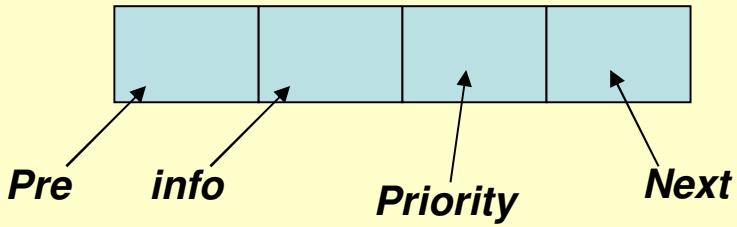
A double ended priority queue is a queue in which one can process the lowest priority element as well as the highest priority element depending upon the requirement.

To implement such a double ended priority queue, following listed methods can be used:

- Double Ended Priority Queue using linked list
- Double Ended Priority Queue using multiple queues
- Double Ended Priority Queue using heap structure

Double Ended Priority Queue using linked list

In the linked representation of Double Ended Priority Queue, each node of the linked list is divided into 4 parts as shown below:



Pre part holds the address of the previous node of the linked list.
Info part holds the element of the queue.
Priority part holds the priority number of the element.
Next part holds the address of next node of linked list.

In linked list representation of double ended priority queue, insertion of an element takes place according to the priority number of the element where as the deletion takes place either from front of linked list or from the end of linked list .

Applications of Priority Queues

- In case of a time sharing systems, where different jobs are to be processed by the same processor, priority queues are used by the operating system to manage the processes. If the operating system of the computer implements the shortest job first policy then in that case, a shortest job will always be processed before the longer job i.e. shortest job will be given priority over the longer job by the operating system. The different job will be given priority over the longer job by the operating system. The different jobs waiting for the processor time will form a priority queue.

Applications of Priority Queues (continued..)

- Priority queue can be used to manage bandwidth on transmission line from a network router. In case of limited bandwidth, all other queues can be halted to send the traffic from the highest priority queue.
- Priority queue can be used in Huffman coding that requires one to repeatedly obtain the two lowest-frequency trees. A priority queue makes this efficient.
- Priority queue has its application in event driven simulation, numerically computation, data compression, graph searching, computational number theory, artificial intelligence, discrete optimization etc.

Applications of Queues

- Queue is used to access the shared resourses e.g. printer queues
- Queue is used as buffer between the fast processor and slow input/output devices.
- Queue is used to implement multiprogramming concepts.
- Queue is used by operating systems for process management.
- Queue can be used as components for the other data structures.
- Queue is used as buffer in leaky bucket algorithm to control the flow of data of the router.