

Webpage

WWWQueuesIdatostructures.com

HOME INFORMATION ON QUEUE LOG IN SIGN IN

remove

insert

remove

insert

remove

insert

remove

insert

remove

insert

remove

insert

QUEUE



This Website will make you understand the concept of Queues

The screenshot shows a web browser window with the address bar displaying 'WWW.Queues\datastructures.com/InformationOnQueue'. The page has a dark gray background. At the top, there is a navigation bar with links: HOME, INFORMATION ON QUEUE, LOG IN, and SIGN IN. The main content area features the title 'QUEUES' in large yellow letters. Below the title, a paragraph in orange text explains that a queue is an abstract data type or a linear data structure, similar to a stack, but with insertion at one end (REAR/tail) and removal at the other (FRONT/head). At the bottom, there are two sections: 'TYPES OF QUEUES' and 'BASIC FEATURES OF QUEUES', each preceded by a circular arrow icon.

Webpage

WWW.Queues\datastructures.com/InformationOnQueue

HOME INFORMATION ON QUEUE LOG IN SIGN IN

QUEUES

Queue is also an abstract data type or a linear data structure, just like stack data structure, in which the first element is inserted from one end called the REAR(also called tail), and the removal of existing element takes place from the other end called as FRONT(also called head).

➡ TYPES OF QUEUES

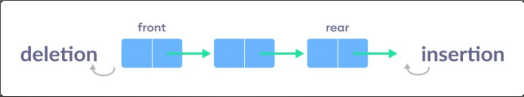
➡ BASIC FEATURES OF QUEUES

[HOME](#) [INFORMATION ON QUEUE](#) [LOG IN](#) [SIGN IN](#)

TYPES OF QUEUES

1: Simple Queue

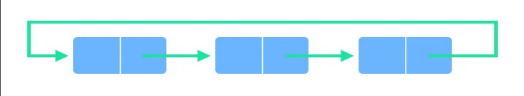
In a simple queue, insertion takes place at the rear and removal occurs at the front. It strictly follows the FIFO (First in First out) rule.



The diagram illustrates a simple queue with three nodes, each represented as a blue rectangle divided into two parts. The first node is labeled 'deletion' with a curved arrow pointing to its left side. The second node is labeled 'front' above it. The third node is labeled 'rear' above it. Green arrows connect the right side of the first node to the left side of the second, and the right side of the second to the left side of the third. A green arrow labeled 'insertion' with a curved arrow pointing to its right side originates from the right side of the third node.

2: Circular Queue

In a circular queue, the last element points to the first element making a circular link.

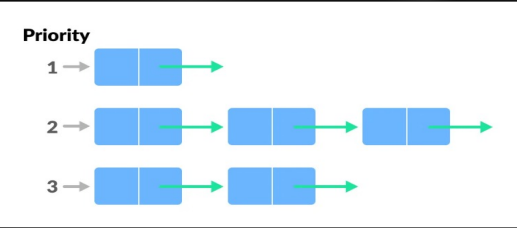


The diagram shows a circular queue with three nodes, each a blue rectangle divided into two parts. Green arrows connect the right side of the first node to the left side of the second, the right side of the second to the left side of the third, and the right side of the third back to the left side of the first, forming a continuous loop.

The main advantage of a circular queue over a simple queue is better memory utilization. If the last position is full and the first position is empty, we can insert an element in the first position. This action is not possible in a simple queue.



3: Priority Queue


A priority queue is a special type of queue in which each element is associated with a priority and is served according to its priority. If elements with the same priority occur, they are served according to their order in the queue. Insertion occurs based on the arrival of the values and removal occurs based on priority.



The diagram illustrates a priority queue with three levels of priority, labeled 1, 2, and 3 on the left. Each level has a green arrow pointing to a blue rectangle divided into two parts. Level 1 has one node. Level 2 has two nodes. Level 3 has two nodes. Green arrows connect the right side of the first node of level 2 to the left side of the second node of level 2, and the right side of the first node of level 3 to the left side of the second node of level 3. Each node also has a green arrow pointing to its right side, indicating the flow of elements within each priority level.

Webpage



HOME INFORMATION ON QUEUE → LOG IN → SIGN IN →

BASIC FEATURES OF QUEUES

- Like stack, queue is also an ordered list of elements of similar data types.
- Queue is a FIFO(First in First Out) structure.
- Once a new element is inserted into the Queue, all the elements inserted before the new element in the queue must be removed, to remove the new element.

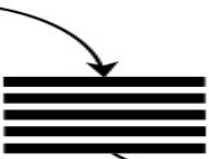
HOME

INFORMATION ON QUEUE

LOG IN

SIGN IN

Queue:




First in, first out

Log IN

Sign IN



Email

 Enter Email

Password

 Enter Password

LOG IN

10	20	30					
0	1	2	3	4	5	6	7
							
Front		Rear					Tutorial4us.com

6

enqueue

tail

5

4

3

head


2

dequeue

1

ORDER

PICK UP



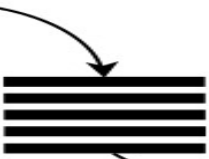
HOME

INFORMATION ON QUEUE

LOG IN

SIGN IN

Queue:




First in, first out


Log IN

Sign IN



Email

 Enter Email

Password

 Enter Password

LOG IN

10	20	30					
0	1	2	3	4	5	6	7
							
Front		Rear					Tutorial4us.com

6

enqueue

tail

5

4

3

head


2

dequeue

1

ORDER

PICK UP



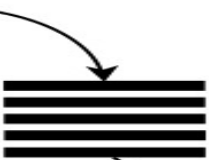
HOME

INFORMATION ON QUEUE

LOG IN

SIGN IN

Queue:



First in, first out

Log IN

Sign IN

UserName

Enter name

Email

Enter Email

Password

Enter Password

Sign IN

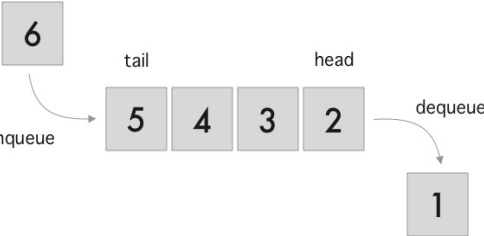
Admin Login


10	20	30					
0	1	2	3	4	5	6	7

Front

Rear

Tutorial4us.com









HOME

INFORMATION ON QUEUE

LOG IN

SIGN IN

QUEUE IMPLEMENTATION PAGE

Enter the Number of Elements

10

Disabled

Reset Queue

Enter the Element to Insert

Insert Element

Remove Queue

FRONT

REAR

FRONT :-

0

REAR :-

3

1

1

1

1


Webpage

WWW.Queuees\datastructures.com/Login/QueueImplementation


HOME INFORMATION ON QUEUE LOG IN LOGIN/

ABOUT US


This website aims to provide user friendly platfrom for the user to learn about Queue data structure




PIYUSH KUMAR SINGH
PES1PG21CA056




PRAKHAR PANDEY
PES1PG21CA058




VIGNESH U.
PES1PG21CA098



PREMALATHA H M



TAMAL DEY



Dr. LEKHA ACHUTH
MENTOR