DATA STRUCTURES AND ALGORITHMS

Queue Data Structure

By
Zainab Malik

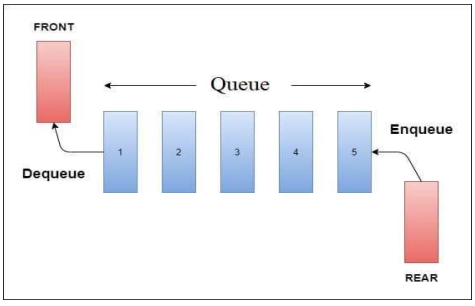
Content

- Introduction to Queue Data Structures
  - Properties of a Queue
  - Operations of Queue
  - Applications of Queue

Queue

- Queue is a linear data structure in which elements are added from an end i.e. **rear**, and removed from another end that is known as the **front**.
- This two end entry and removal ensures the first-in-first-out (FIFO) or last-in-last-out (LILO) order of insertion and deletion.
- By convention insertion and deletion in queue are termed as ENQUEUE and DEQUEUE, respectively.

Queue



5

Operations of Queue

- The common operations of queue are as follow:
  - enqueue()
  - dequeue()
  - isEmpty()
  - isFull()
  - frontValue()
  - rearValue()

6

Operations of Queue-Enqueue(item)

Enqueue (queue, item)

1. If queue is already full:
2. Display an error of “overflow”
3. If queue is empty and this is the first item to be inserted in that queue
4. Increment rear and front both
5. Insert item at rear index
6. Otherwise:
7. Increment rear
8. Insert item at rear index

7

QUEUE- Enqueue Operation

Operation	Rear	front	0	1	2	3	4	5
-	-1	-1						
Enqueue(a)	0	0	a					
Enqueue(b)	1	0	a	b				
Enqueue(c)	2	0	a	b	c			
Enqueue(d)	3	0	a	b	c	d		
Enqueue(e)	4	0	a	b	c	d	e	
Enqueue(f)	5	0	a	b	c	d	e	f
Error Enqueue(g)	5	0	a	b	c	d	e	f

8

Operations of Queue-Dequeue()

Dequeue (queue):

1. If Queue is already empty:
2. Display an error of “underflow”
3. If there is only one element in the queue
4. Save value of front index in a variable “Item”
5. Set front and rear both to -1
6. Return Item
7. Otherwise:
8. Save value of front index in a variable “Item”
9. Increment front
10. Return Item

9

QUEUE- Dequeue Operation

Operation	Rear	front	0	1	2	3	4	5
-	4	0	a	b	c	d	e	
Dequeue()	4	1		b	c	d	e	
Dequeue()	4	2			c	d	e	
Enqueue(f)	5	2			c	d	e	f
Dequeue()	5	3				d	e	f
Dequeue()	5	4					e	f
Dequeue()	5	5						f
Dequeue()	-1	-1						

10

Operations of Queue-isFull()

isFull():

1. If rear is at size-1:
2. Return true
3. Otherwise:
4. Return false

Operation	Rear	front	0	1	2	3	4	5
isFull()	5	0	a	b	c	d	e	f

True

Operation	Rear	front	0	1	2	3	4	5
isFull()	2	0	a	b	c			

False

Operation	Rear	front	0	1	2	3	4	5
isFull()	5	3				d	e	f

True

11

Operations of Queue-isEmpty()

isEmpty():

1. If rear and front are at -1:
2. Return true
3. Otherwise:
4. Return false

Operation	Rear	front	0	1	2	3	4	5
isEmpty()	-1	-1						

True

Operation	Rear	front	0	1	2	3	4	5
isEmpty()	2	0	a	b	c			

False

12

Operations of Queue-frontValue()

frontValue():

1. If rear and front are at -1:
2. Display error “underflow”
3. Otherwise:
4. Return value at front index

Operation	Rear	front	0	1	2	3	4	5
frontValue()	-1	-1						

Error

Operation	Rear	front	0	1	2	3	4	5
frontValue()	2	0	a	b	c			

a

13

Operations of Stack-rearValue()

rearValue():

1. If rear and front are at -1:
2. Display error “underflow”
3. Otherwise:
4. Return value at rear index

Operation
rearValue()

Rear	front
-1	-1

0	1	2	3	4	5

Error

Operation
rearValue()

Rear	front
2	0

0	1	2	3	4	5
a	b	c			

c

14

Applications of Queue

- It is used in all those application where FIFO/LILO order is mandatory.
- It is used for scheduling purpose
- It can be used for buffering of data packets, where order of packets must be maintained

15

Thank You