15CSE374
INTRODUCTION TO DATA STRUCTURES AND ALGORITHMS

Sarath tv

Last Lecture

- List
- Array
- Stack as ADT.
- Implementation of stack

Queue

- A data structure used for storing data.
- Order in which data arrives is important.
- An ordered list in which insertions are done at one end and deletion are done at the other end.
- FIFO/LILO

Queue

Queues are data structures that, like the stack, have restrictions on where you can add and remove elements. To understand a queue, think of a cafeteria line: the person at the front is served first, and people are added to the line at the back.

Thus, the first person in line is served first, and the last person is served last. This can be abbreviated to **F**irst **I**n, **F**irst **O**ut (**FIFO**).

The cafeteria line is one type of queue. *Queues are often used* in programming networks, operating systems, and other situations in which *many different processes must share resources* such as CPU time.

One bit of terminology: the addition of an element to a queue is known as an **enqueue**, and removing an element from the queue is known as a **dequeue**.

Although the concept may be simple, **programming a queue is not as simple as programming a stack**.

Let's go back to the example of the cafeteria line. Let's say one person leaves the line. Then what? Everyone in line must step forward, left?

Queue Interfaces

- Enqueue- Add new element into the queue.
- Dequeue- Remove the first element in the queue.
- isEmpty
- Peek.
- Size

General points on implementation of Queue.

Two pointers: Front and Rear.

Enqueue – Operation adds elements at the rear of the queue provided there is space for adding data.

Dequeue –Operation that removes the elements at the front of the queue provided the queue is not empty.

Underflow Condition

Overflow Condition

ARRAY REPRESENTATION OF QUEUES

12	9	7	18	14	36				
0	1	2	3	4	5	6	7	8	9

Queue FRONT = 0 and REAR = 5.

12	9	7	18	14	36	45			
0	1	2	3	4	5	6	7	8	9

Queue after insertion of a new element

FRONT = 0 and REAR = 6.

	9	7	18	14	36	45			
0	1	2	3	4	5	6	7	8	9

Queue after deletion of an element

FRONT = 1 and REAR = 6.

enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue().

Applications

- OS schedule jobs in the order of arrival.
- Waiting times of customers at call centers.
- Simulation of real world queues (lines at ticket counter).

implementations

- List(linked list based, array based…)
- collections.deque
- queue.Queue

THANK YOU!!!!!