

DATA STRUCTURE

□ INTRODUCTION TO QUEUE (FIFO)

EMAN T.MAHDI
COLLEGE OF C.S. &I.T.

OVER VIEW

- QUEUE
- **INITIALIZE QUEUE**
- **QUEUE OPERATIONS**
- **QUEUE ALGORITHMS**

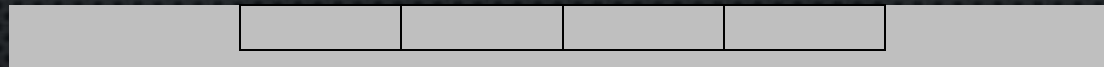
QUEUE

A QUEUE IS AN ORDERED COLLECTION OF ITEMS FROM WHICH ITEMS MAY BE DELETED AT ONE END (COLLECT THE FRONT OF THE QUEUE) AND INTO WHICH ITEMS MAY BE INSERTED AT THE OTHER END (CALLED THE REARER OF THE QUEUE).

INITIALIZE QUEUE

THE FIRST ELEMENT INSERTED INTO THE QUEUE IS THE FIRST ELEMENT TO BE REMOVED. FOR THIS REASON A QUEUE IS SOMETIMES CALLED A **FIFO** (FIRST-IN FIRST-OUT) LIST AS OPPOSED TO THE STACK, WHICH IS A LIFO (LAST-IN FIRST-OUT).

Initial queue

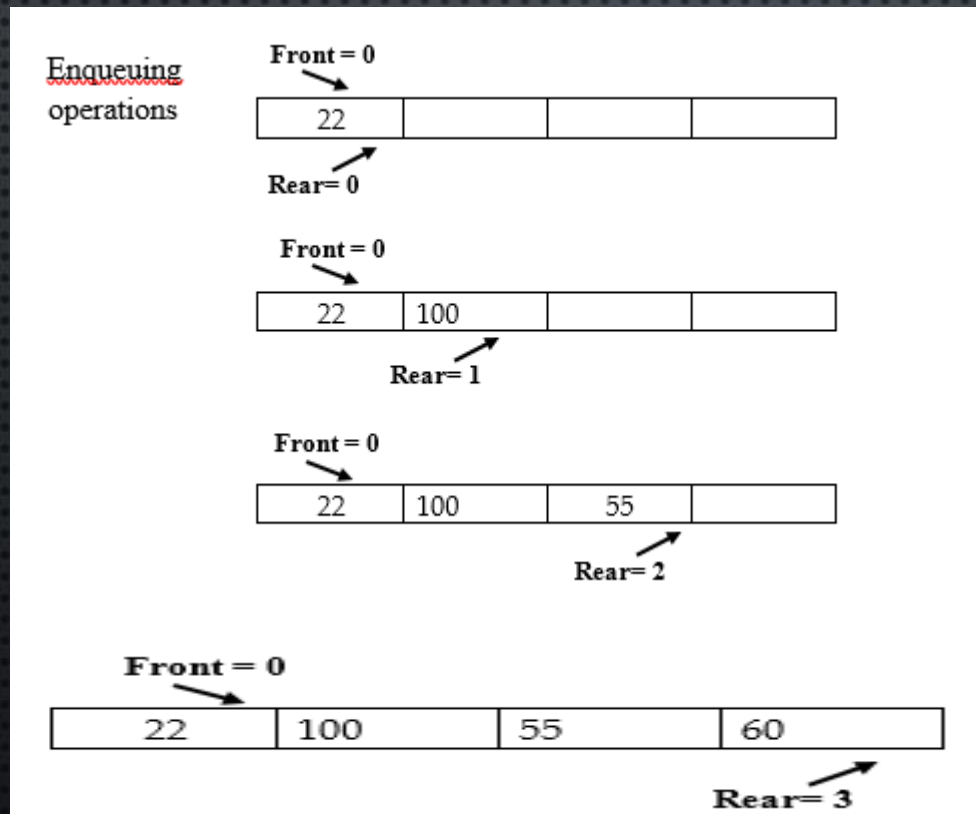


Front = -1

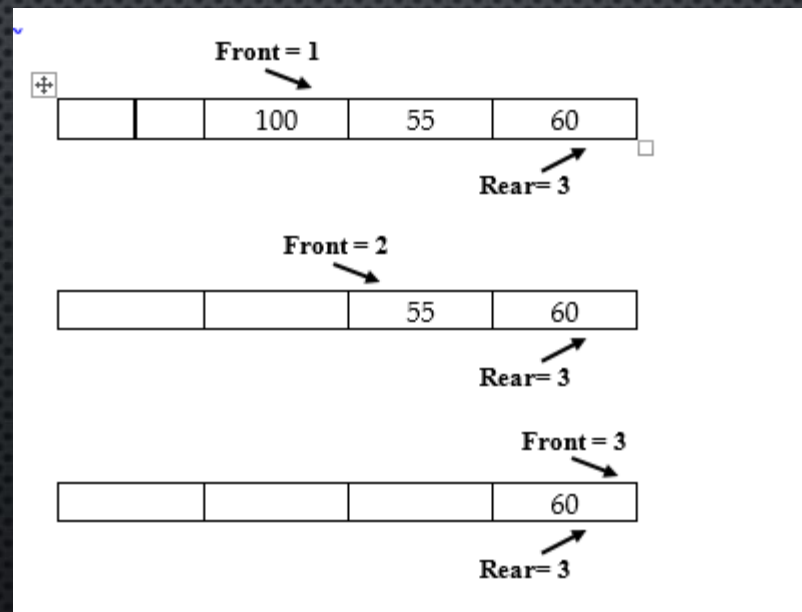
Rear = -1

QUEUE OPERATIONS

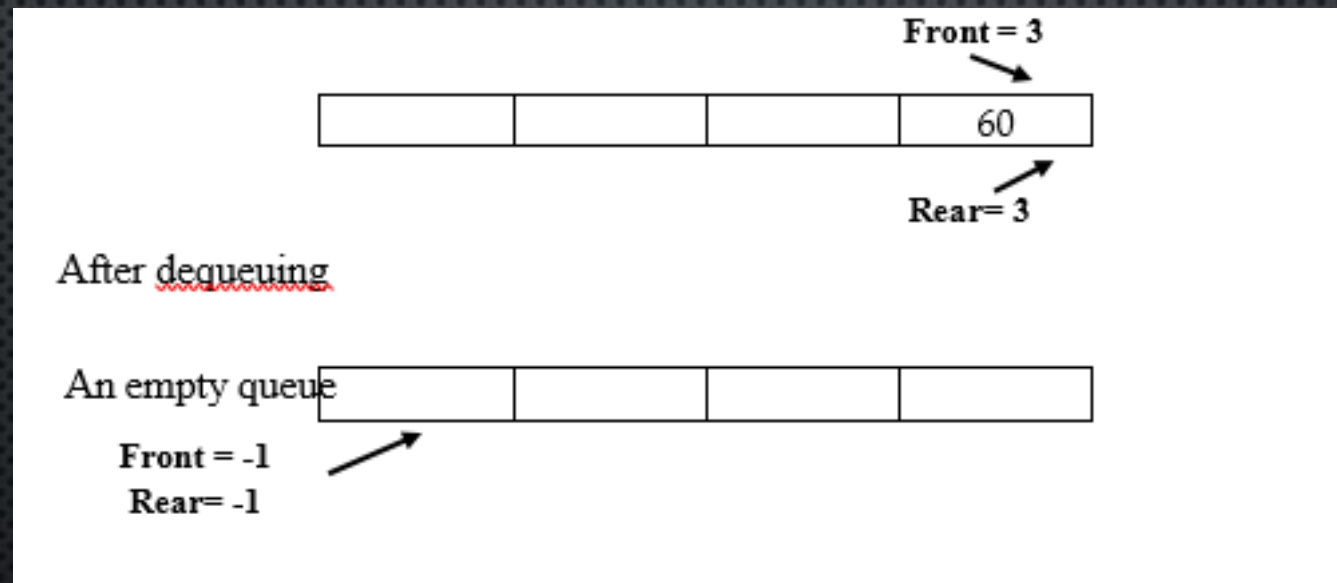
- THE TWO PRIMARY QUEUE OPERATIONS ARE **ENQUEUEING** AND **DEQUEUEING**.
- TO ENQUEUE MEANS TO INSERT AN ELEMENT AT THE REAR OF A QUEUE



TO DEQUEUE MEANS TO REMOVE AN ELEMENT FROM THE FRONT OF A QUEUE



AN EMPTY QUEUE CAN BE SIGNIFIED BY SETTING BOTH FRONT AND REAR INDICES TO -1 .



QUEUE ALGORITHMS

- THERE ARE TWO ALGORITHMS TO INSERT (**ENQUEUEING**) AND REMOVE (**DEQUEUEING**) ITEMS INTO AND FROM QUEUE.

- Insert Algorithm (**Enqueueing**)

- [overflow]

- if $R \geq N-1$

- Then Over flow

- [increment Rear]

- Rear = Rear+1

- [insert element]

- Queue[Rear] = New element

- [set Front]

- If Front=-1

- Then Front =0

DELETE ALGORITHM (**DEQUEUEING**)

- [UNDERFLOW]
- IF FRONT= -1
- THEN UNDER FLOW
- [DELETE ELEMENT]
- ELEMENT==QUEUE[FRONT]
- [CHECK EMPTY QUEUE]
- If FRONT =REAR
- THEN FRONT =REAR= -1
- ELSE FRONT=FRONT+1

QUEUE APPLICATIONS

- A WAITING LINE IS A GOOD REAL-LIFE EXAMPLE OF A QUEUE. (A WAITING LINE IN A STORE, AT A SERVICE COUNTER)
- EQUAL-PRIORITY PROCESSES WAITING TO RUN ON A PROCESSOR IN A COMPUTER SYSTEM

QUEUE PROBLEM: FALSE-OVERFLOW ISSUE FIRST

- FOR QUEUE ARRAY SIZE [50] .SUPPOSE 50 CALLS TO ENQUEUE HAVE BEEN MADE, SO NOW THE QUEUE ARRAY IS FULL
- ASSUME 4 CALLS TO DEQUEUE() ARE MADE
- ASSUME A CALL TO ENQUEUE() IS MADE NOW. THE TAIL PART SEEMS TO HAVE NO SPACE, BUT THE FRONT HAS 4 UNUSED SPACES; IF NEVER USED, THEY ARE WASTED.
- **SOLUTION: A CIRCULAR QUEUE**

REFERENCES

- : INTRODUCTION TO ALGORITHMS, 3RD EDITION BY THOMAS H. CORMEN ,CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN
- INTRODUCTION TO ALGORITHMS, 3RD EDITION BY THOMAS H. CORMEN ,CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN
- ELEMENTS OF PROGRAMMING INTERVIEWS IN JAVA: THE INSIDERS' GUIDE, BY ADNAN AZIZ, TSUNG-HSIEN LEE, AMIT PRAKASH
- [HTTPS://GITHUB.COM/CAREERMONK/DATASTRUCTURESANDALGORITHMSMADEEASY](https://github.com/careermonk/DataStructuresAndAlgorithmsMadeEasy)