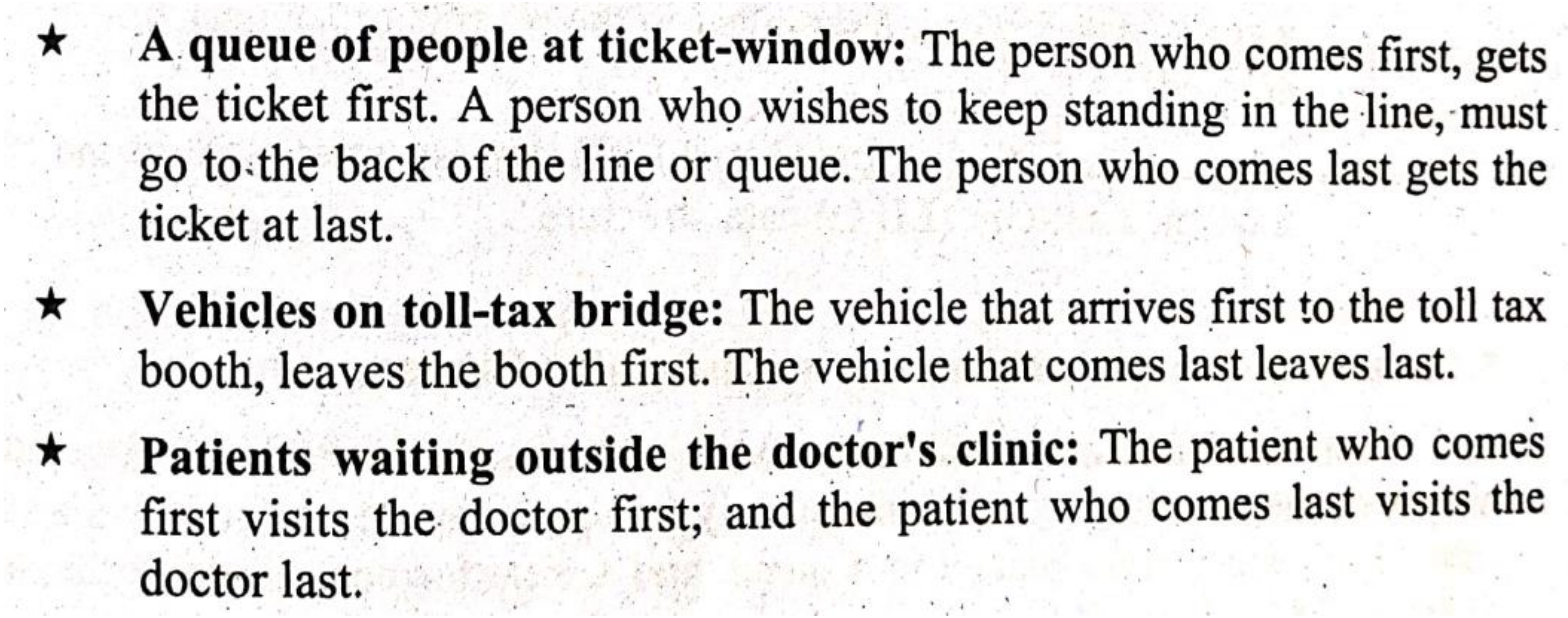
QUEUE

Instructor: Javeria Naz

What is Queue?

- A Queue is a linear structure which follows a particular order in which the operations are performed.
- The order is First In First Out (FIFO).
- A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first.
- The difference between stacks and queues is in removing.
- In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

Cont.

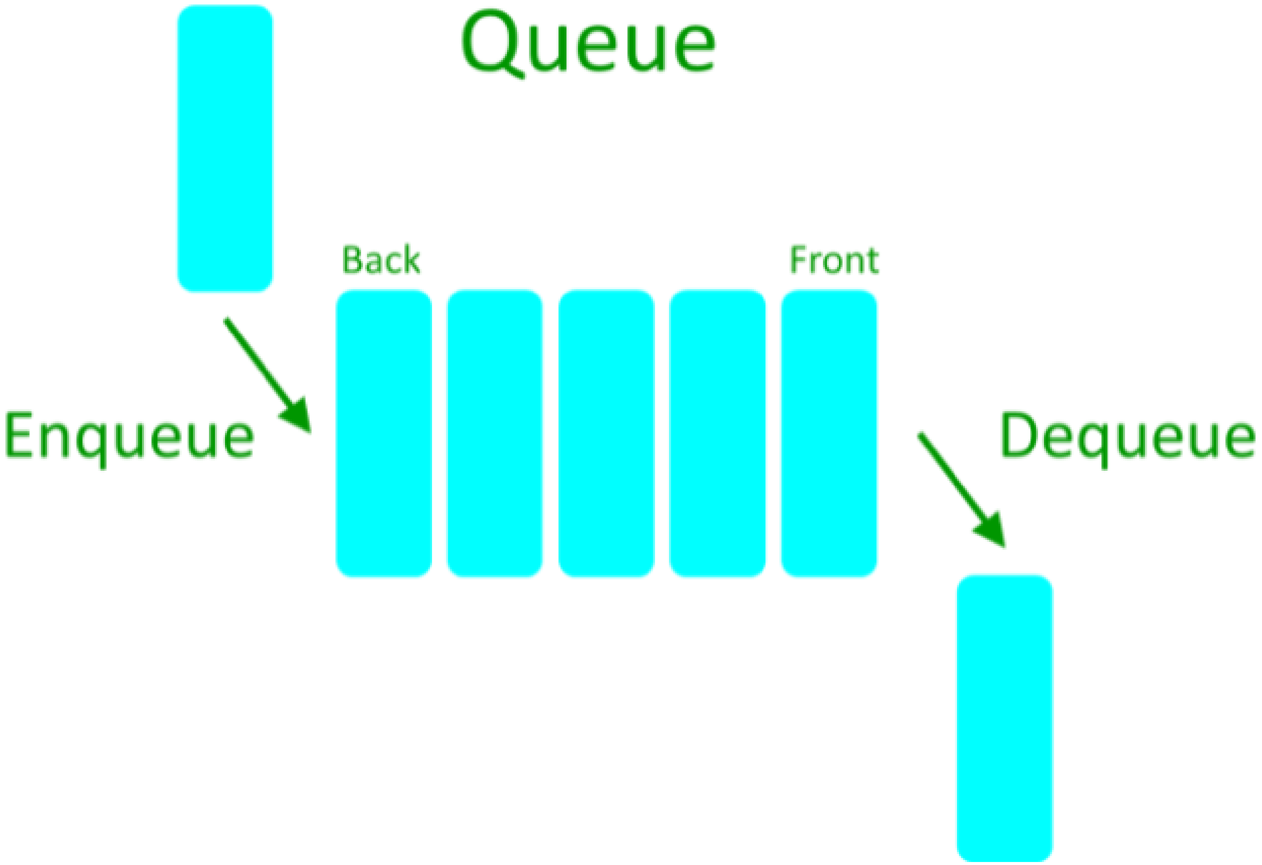


★ **A queue of people at ticket-window:** The person who comes first, gets the ticket first. A person who wishes to keep standing in the line, must go to the back of the line or queue. The person who comes last gets the ticket at last.

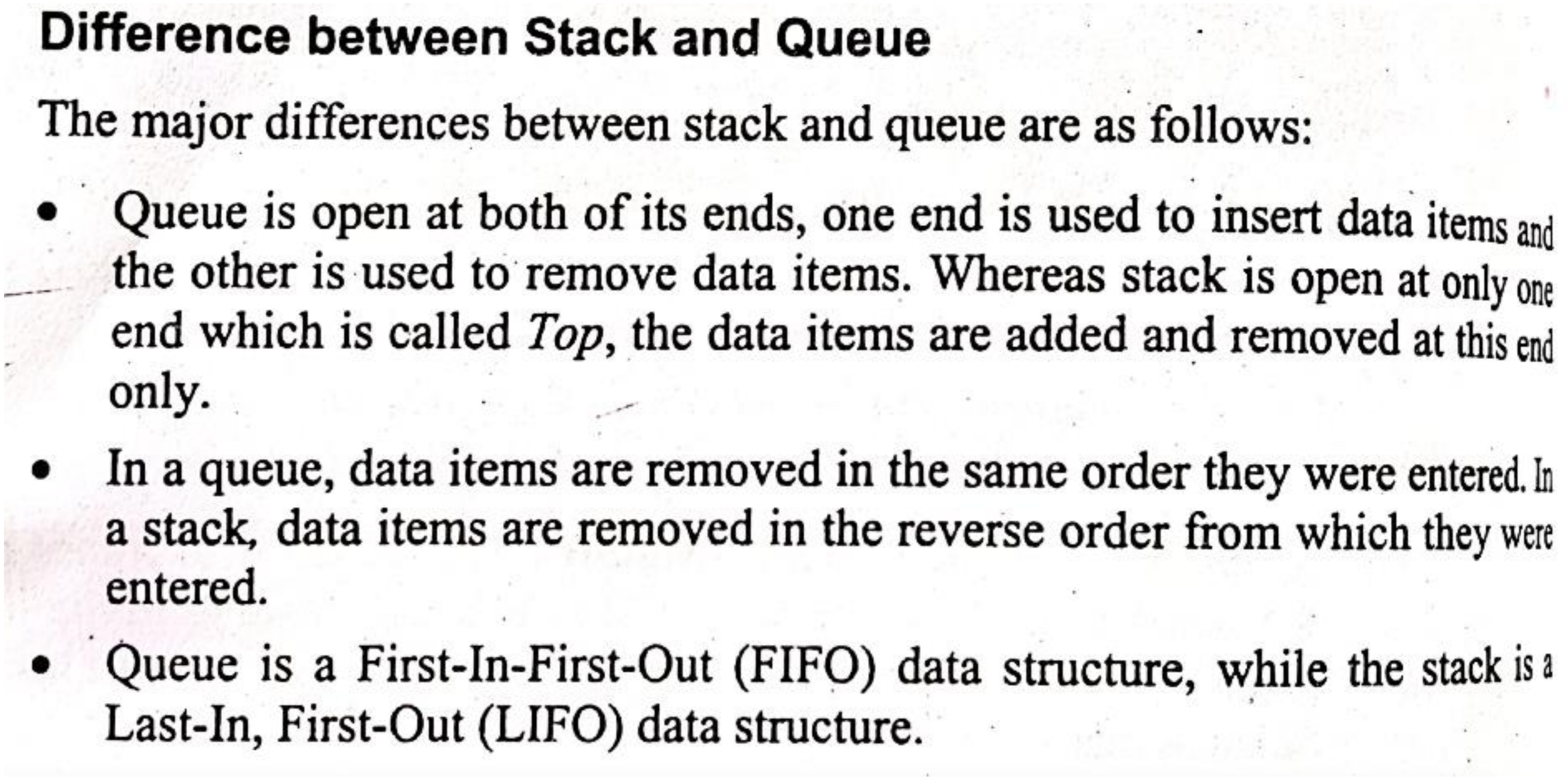
★ **Vehicles on toll-tax bridge:** The vehicle that arrives first to the toll tax booth, leaves the booth first. The vehicle that comes last leaves last.

★ **Patients waiting outside the doctor's clinic:** The patient who comes first visits the doctor first; and the patient who comes last visits the doctor last.

Queue Data Structure



Stack vs Queue



Difference between Stack and Queue

The major differences between stack and queue are as follows:

- Queue is open at both of its ends, one end is used to insert data items and the other is used to remove data items. Whereas stack is open at only one end which is called *Top*, the data items are added and removed at this end only.
- In a queue, data items are removed in the same order they were entered. In a stack, data items are removed in the reverse order from which they were entered.
- Queue is a First-In-First-Out (FIFO) data structure, while the stack is a Last-In, First-Out (LIFO) data structure.

Applications in Computer Systems

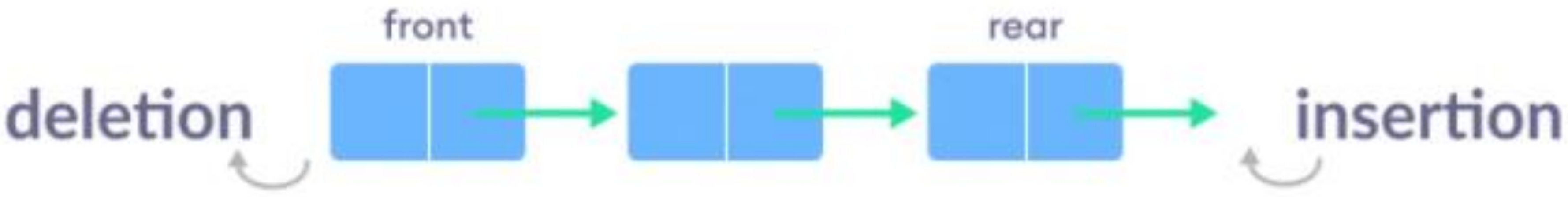
- Queues have many applications in computer systems. Most computers have only a single processor, so the job of only one user can be executed at a time.
- The jobs of the other users are placed in the queue and are performed one after another.
- If the user want to print a document, operating system ques the document and place I special area called Print buffer or print queue.
- The document sent first is printed first a Similarly, network operating system uses queue to process printing requests. The documents are sent to the printer in network environment (which may have single printer).
- The printer print the documents in the order in which they are received.

Types of Queues

- There are four different types of queues:
  - Simple Queue
  - Circular Queue
  - Priority Queue
  - Double Ended Queue

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.



Circular Queue

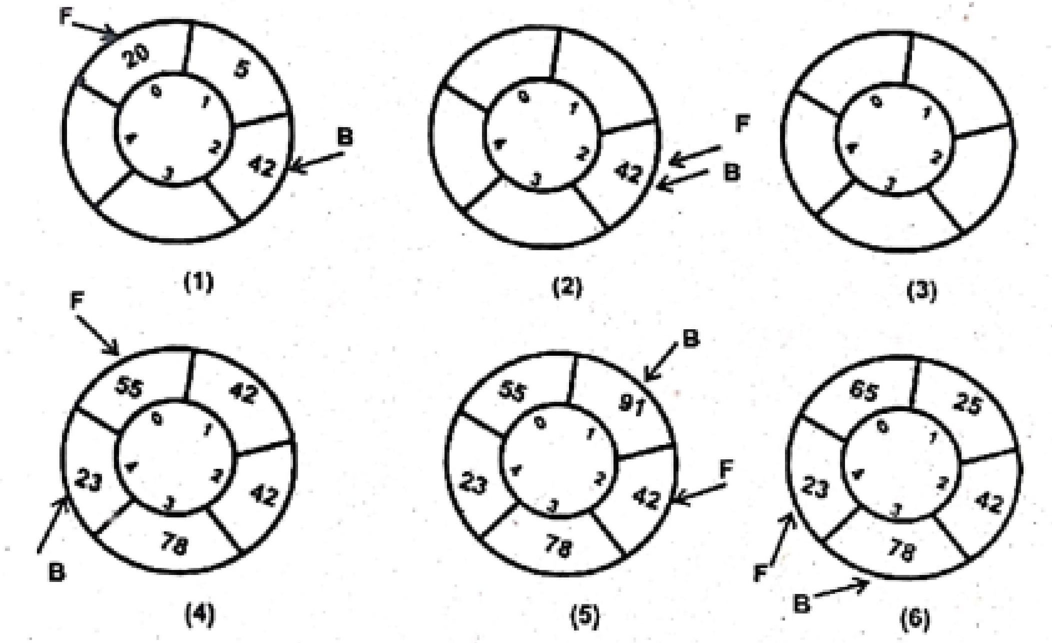
- Like simple queue, there are two basic operations that are performed on circular queue.
- These operations are insertion (enqueue) and deletion (dequeue).
- **Insertion Operation:**
- Insertion operation is used to (add or insert) an item or element into the circular queue. The insertion operation involves the following steps:
  1. Check if the circular queue is full. If it is full, produce an error message "overflow" and exit. A circular queue is declared as full if the value of F is equal to (B+1) % MAX, where MAX indicates the maximum size of array and % is the division operator giving remainder of division.
  2. If circular queue is not full, set the value of B to point to next empty space and then insert new data item into the circular queue. In insertion operation, the value of B is set or updated using the following formula: B = (B+1)% MAX
  3. Check if the value of F is -1; if true then set value of F to 0.
  4. Exit the operation.

Deletion operation

- Deletion operation is used to remove or delete) an item from circular queue. The deletion operation involves the following steps:
  1. Check if the circular queue is empty. If it is empty, produce an error message "underflow" and exit. A circular queue is declared as empty if both F and B are equal to -1.
  2. If the queue is not empty, delete the data item from the location to where F is pointing.
  3. If value of B is equal to F, then set the values of both F and B to **"-1"** otherwise set the value of F using the following formula: F = (F+1) % MAX
  4. Exit the operation

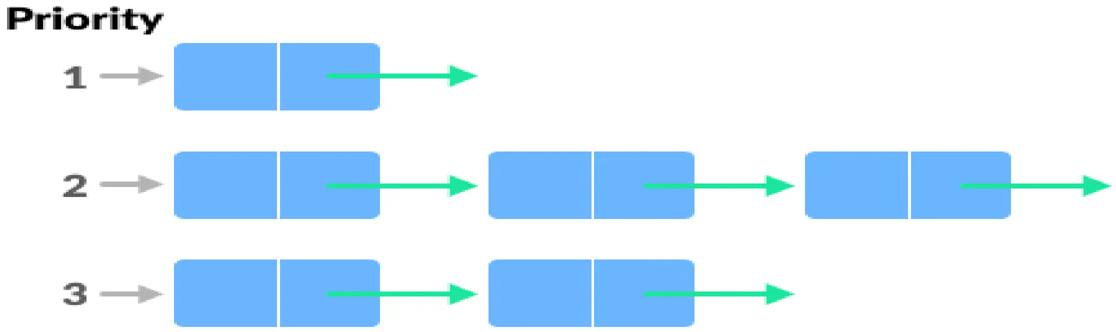
Circular Queue

- A circular queue (4), (5), and (6) show full circular queues with different values of F circular queue is declared as full queue if the value of F is equal to % MAX (MAX means maximum size of array).
- In figure (4), value of B is 4, F is calculated as: 4+1 % 5= 0.
- In figure (5), value of B is 1, F is calculated as: 1+1 % 5 = 2.
- In figure (6), value of B is 3, F is calculated as: 3+1 % 5 = 4.



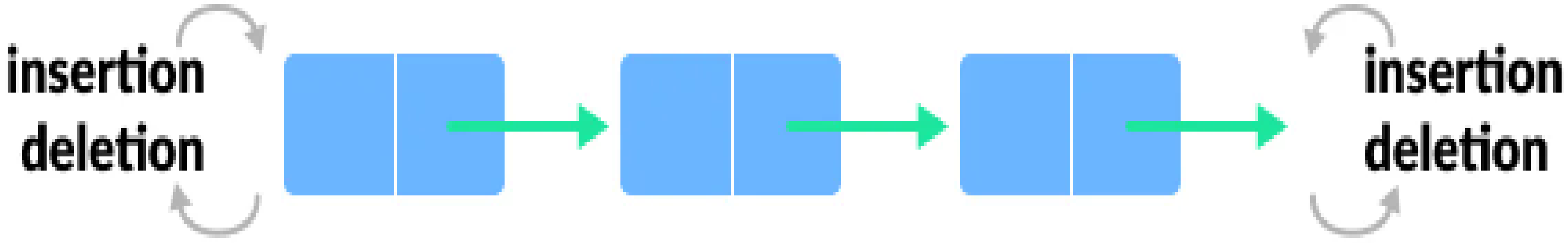
Priority Queue

- 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.

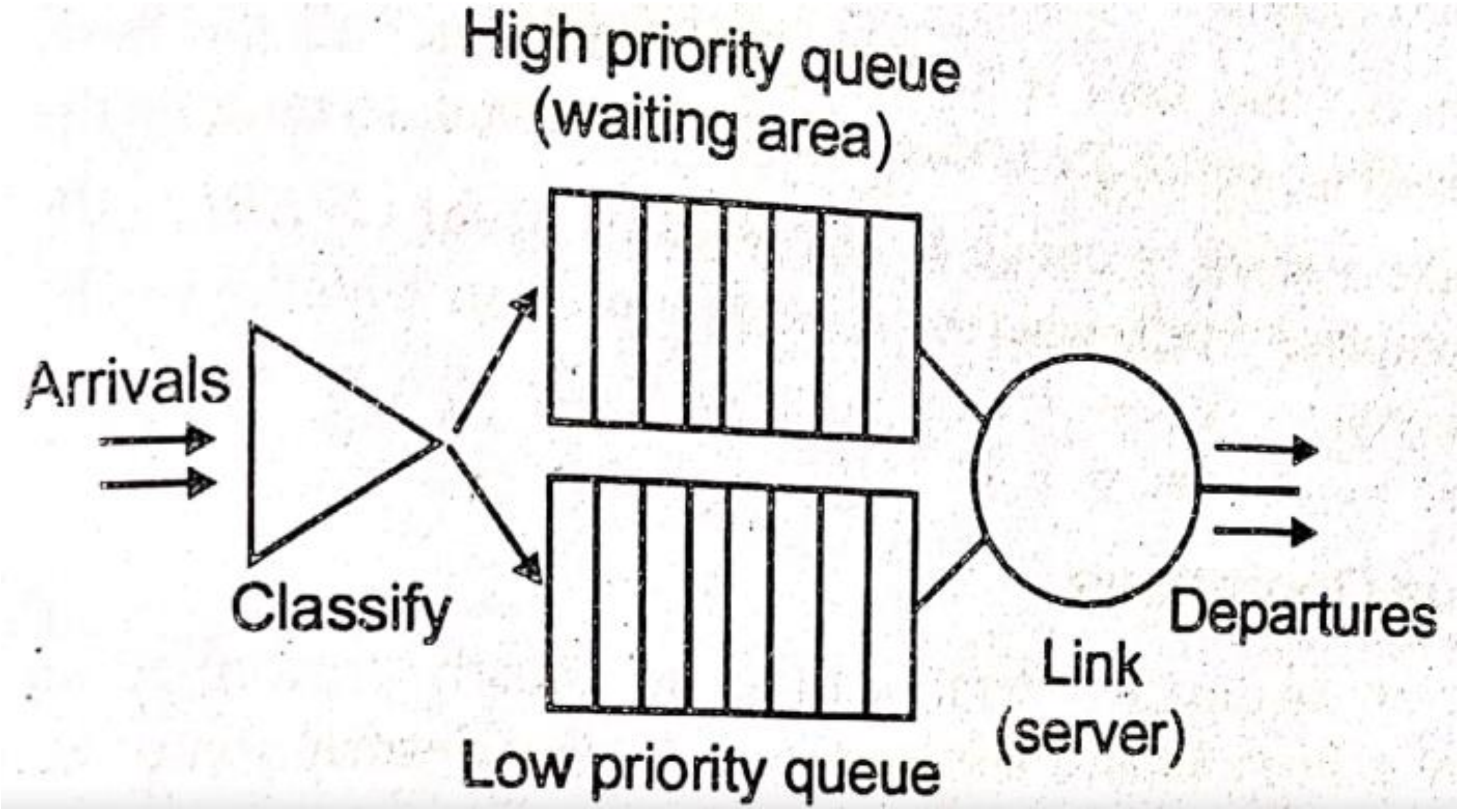


Deque (Double Ended Queue)

- In a double ended queue, insertion and removal of elements can be performed from either from the front or rear.
- Thus, it does not follow the FIFO (First In First Out) rule.



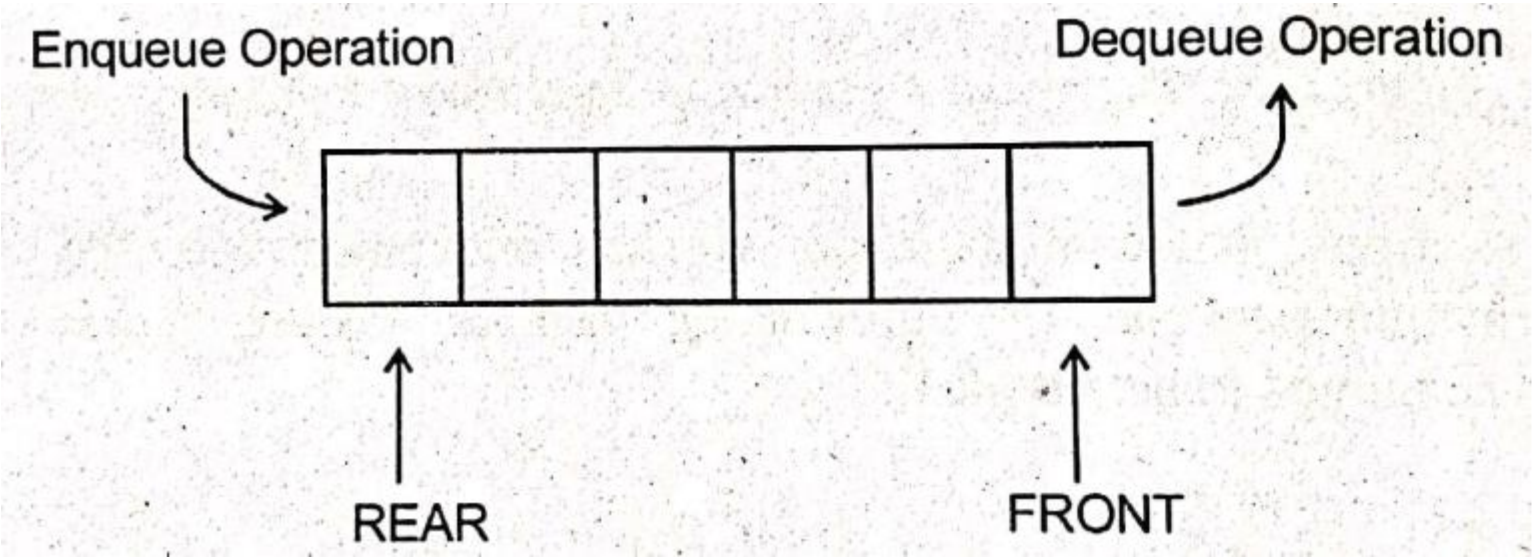
Priority Queue in Multiprogramming System



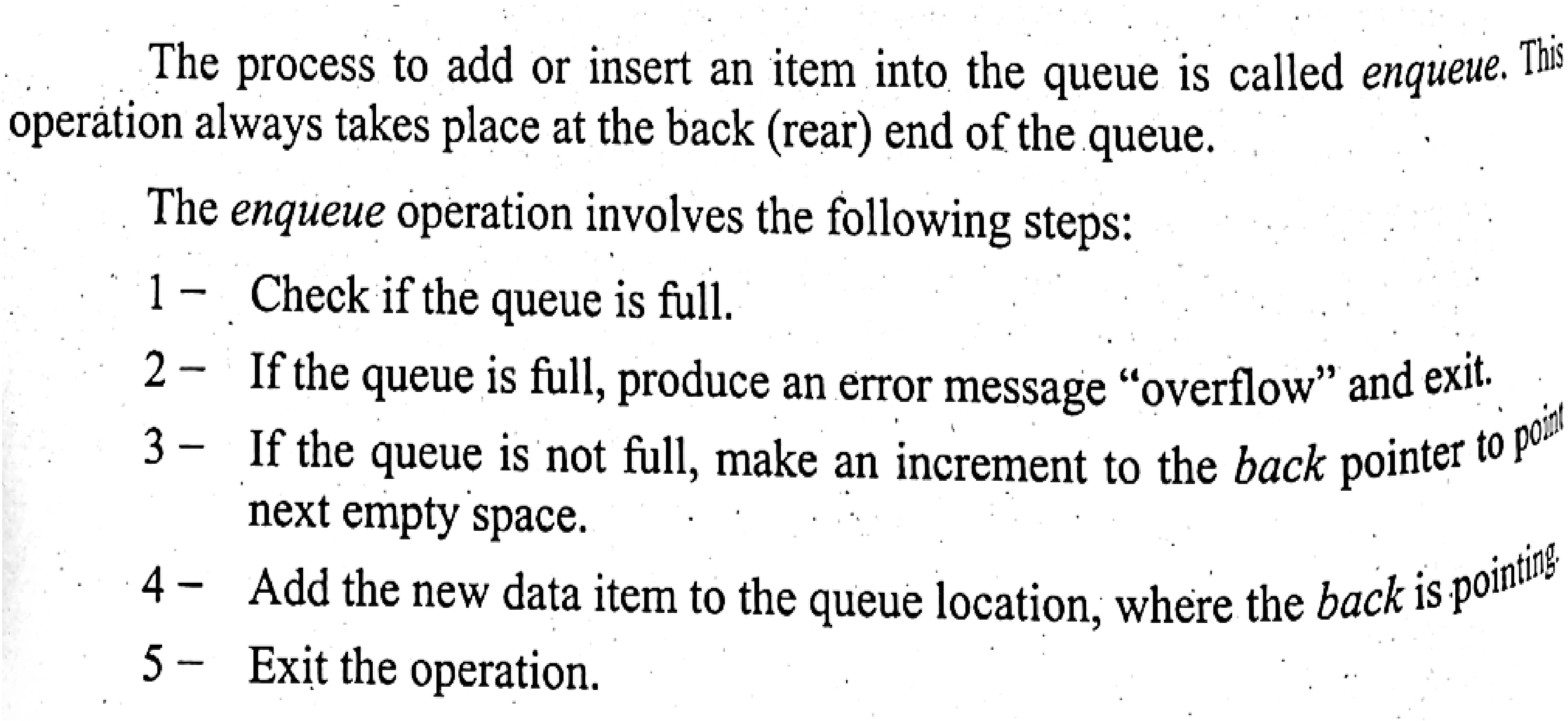
Features of Queue

1. Like stack, queue is also an ordered list of items or elements with data types.
2. Queue is a FIFO data structure (i.e. items are accessed in First Out) manner.
3. Once a new item is inserted into the queue, all the items inserted the new item in the queue must be removed, to remove the new item.
4. Like a stack, a queue is also said to be in OVERFLOW state when completely full, whereas it is said to be in UNDERFLOW state when it is empty.

- There are two basic operations of queue called enqueuer and dequeuer.



Enqueue Operation



The process to add or insert an item into the queue is called *enqueue*. This operation always takes place at the back (rear) end of the queue.

The *enqueue* operation involves the following steps:

1 – Check if the queue is full.

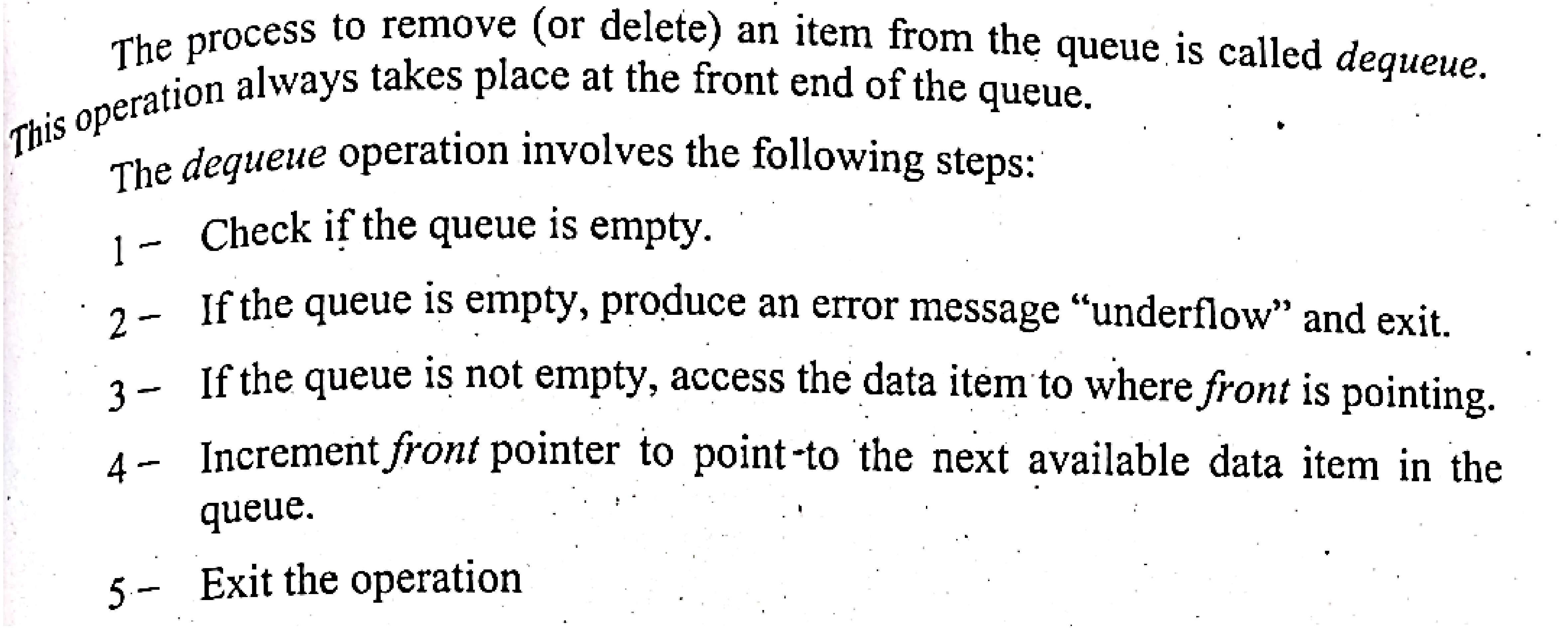
2 – If the queue is full, produce an error message "overflow" and exit.

3 – If the queue is not full, make an increment to the *back* pointer to point next empty space.

4 – Add the new data item to the queue location, where the *back* is pointing.

5 – Exit the operation.

Dequeue Operation



The process to remove (or delete) an item from the queue is called *dequeue*. *This operation* always takes place at the front end of the queue.

The *dequeue* operation involves the following steps:

1 – Check if the queue is empty.

2 – If the queue is empty, produce an error message "underflow" and exit.

3 – If the queue is not empty, access the data item to where *front* is pointing.

4 – Increment *front* pointer to point-to the next available data item in the queue.

5 – Exit the operation

Priority Queue

- Priority Queue is an extension of queue with following properties.
  - An element with high priority is dequeued before an element with low priority.
  - If two elements have the same priority, they are served according to their order in the queue.
  - Every item has a priority associated with it.

A typical priority queue supports following operations:

- ✓ **insert(item, priority):** Inserts an item with given priority.
- ✓ **getHighestPriority():** Returns the highest priority item.
- ✓ **deleteHighestPriority():** Removes the highest priority item.