Intro to data structure & algorithms

by Mariana Makram / ITI-sohag

Stack - LIFO

A Stack is a linear data structure that follows the LIFO (Last-In-First-Out) principle.

A stack can be defined as a container in which insertion and deletion can be done from the one end known as the top of the stack.

Standard Stack Operations

push()

insert an element in a stack

pop()

delete an element from the stack

peek()

returns the element at top

display()

print all the elements in the stack.

isEmpty()

determines the stack is empty or not.

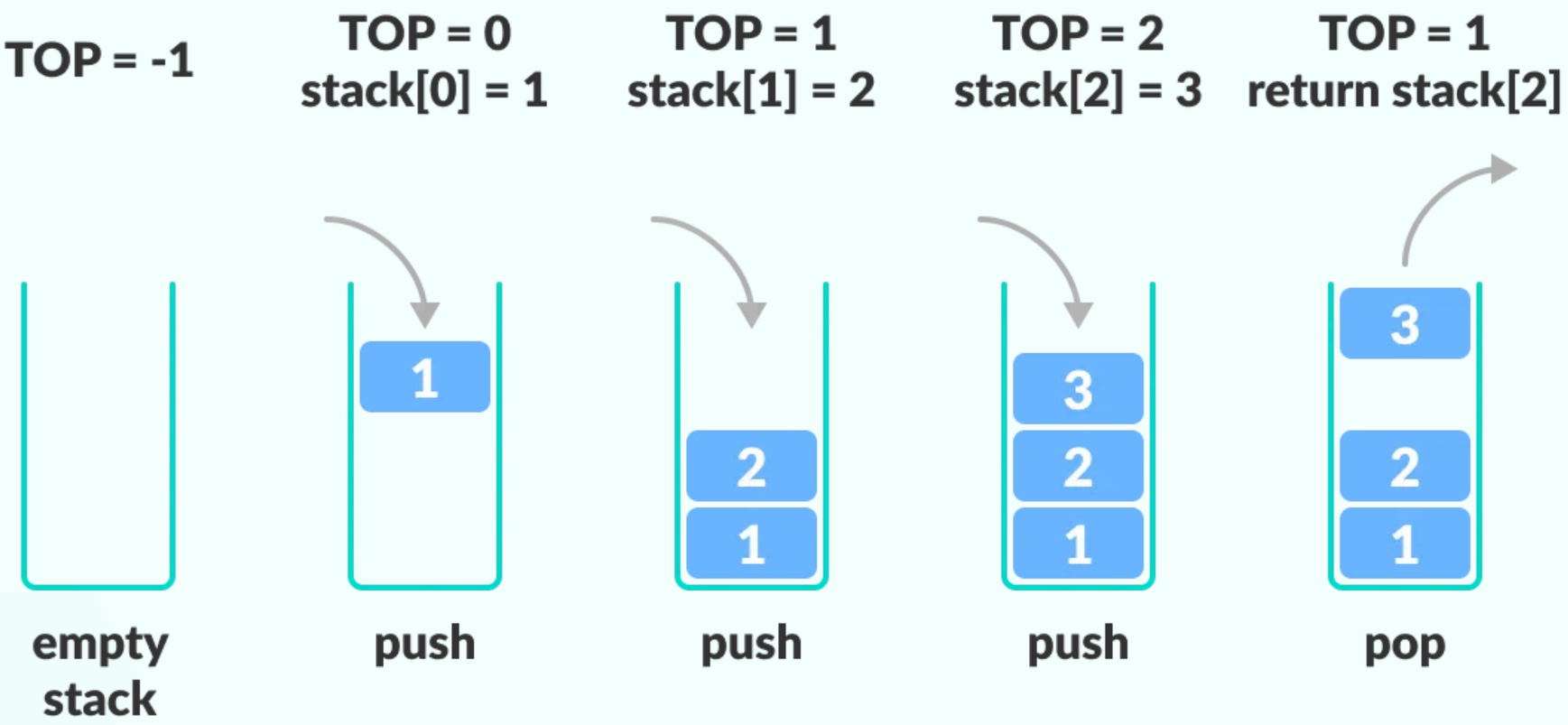
isFull()

determines the stack is full or not.

Implementation stack with array

- Make array behave like a Stack.
- So it should have push() method to add data on top and pop() method to remove data from top.

Implementation stack with array



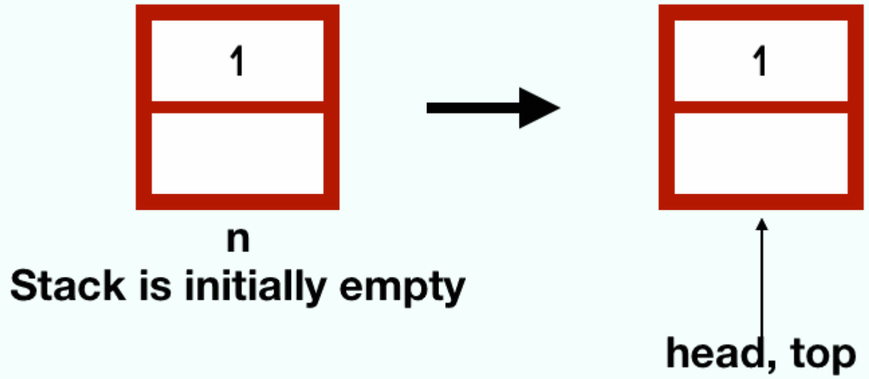
Implementation stack with linked list

- Make a Linked List behave like a Stack.
- So it should have push() method to add data on top and pop() method to remove data from top.

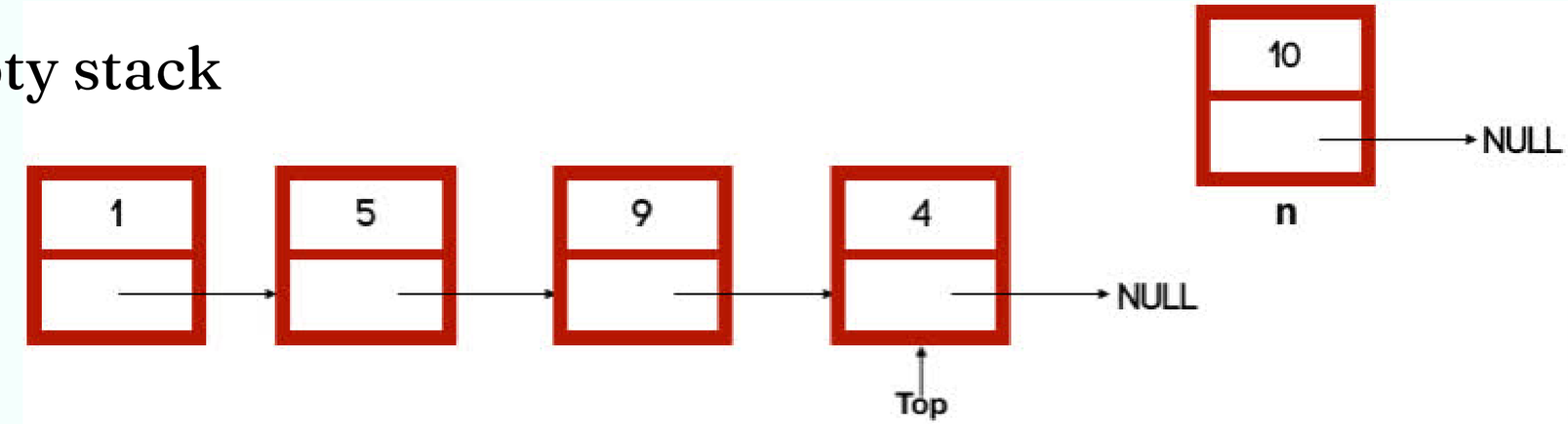
Implementation stack with linked list

push( )

Push on an empty stack

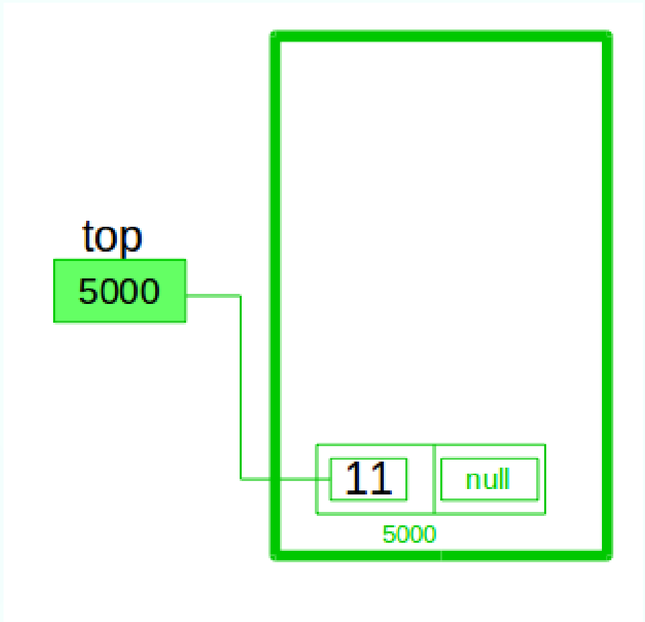


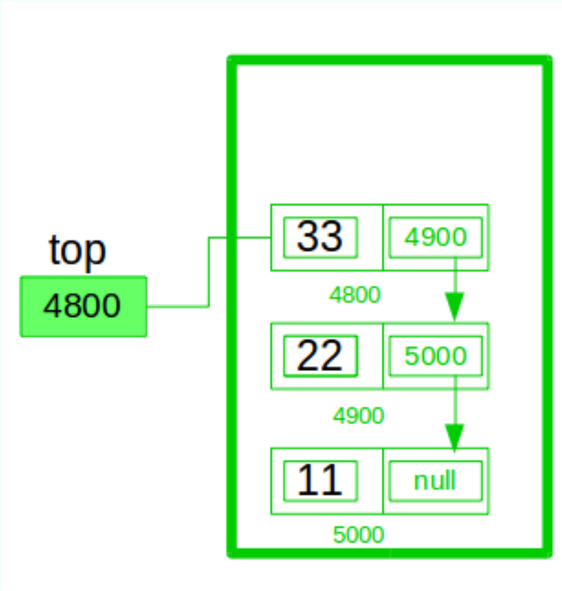
Push on a non-empty stack

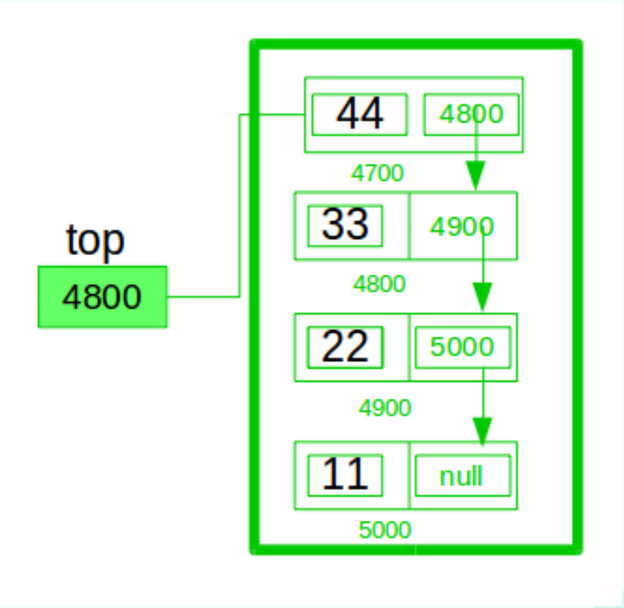


Implementation stack with linked list

push( )





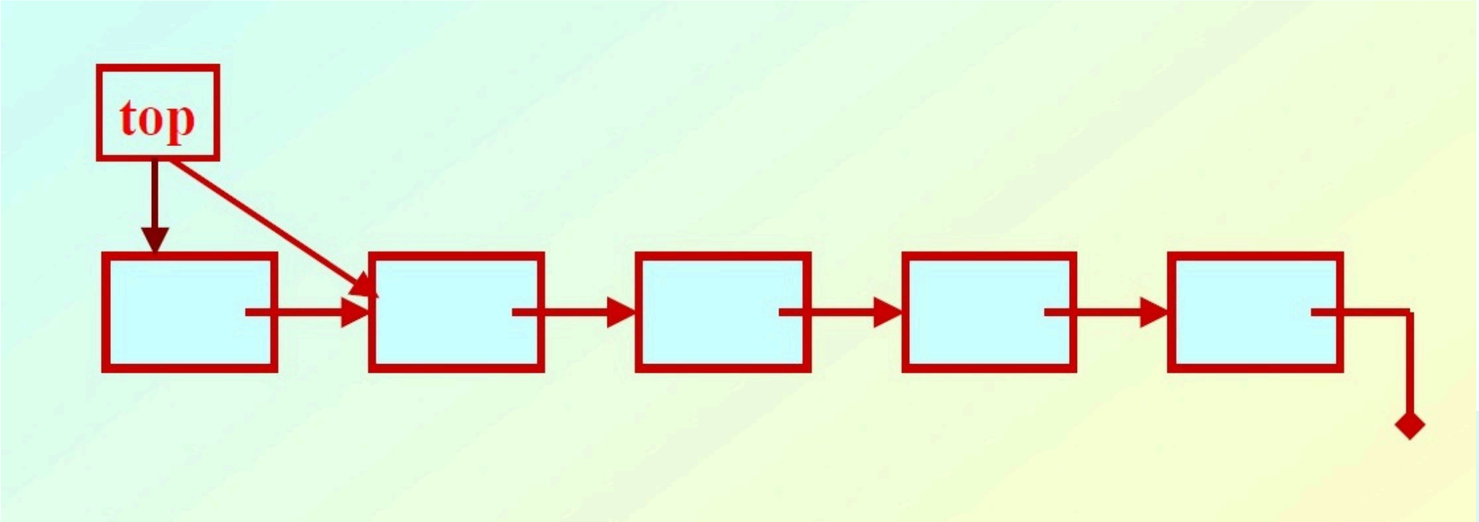


Implementation stack with linked list

pop()

Pop from an empty stack --> None.

Pop from a non-empty stack --> value

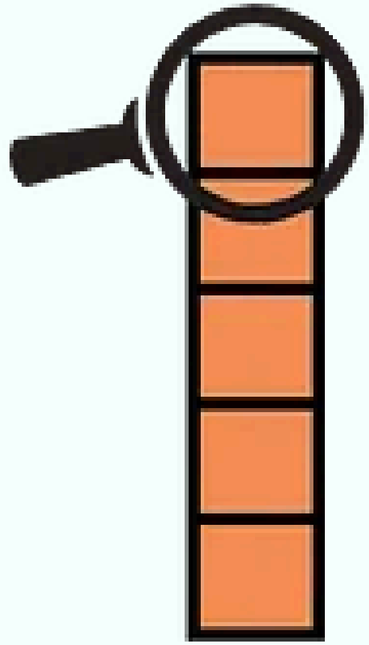


Implementation stack with linked list

peek( )

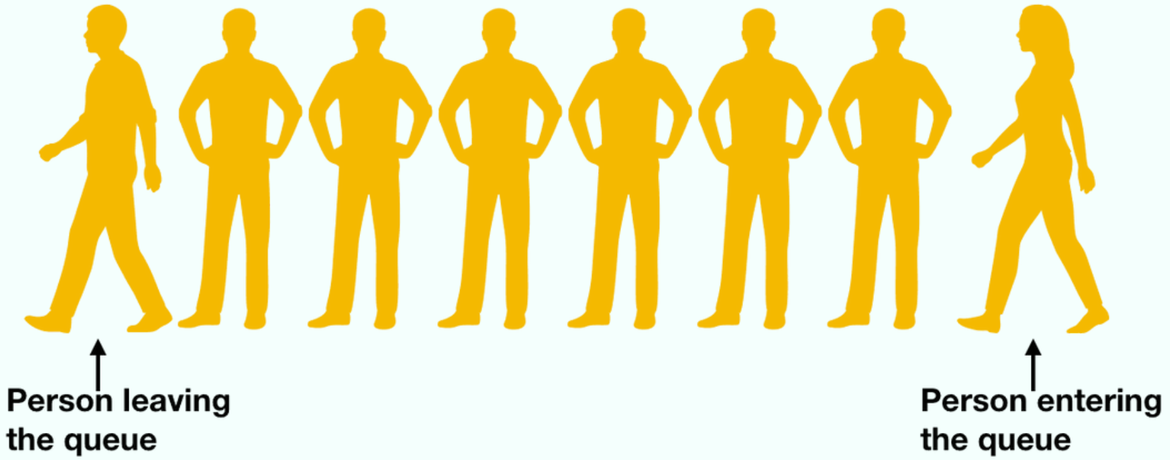
Peek on an empty stack --> None.

Peek on a non-empty stack --> value.



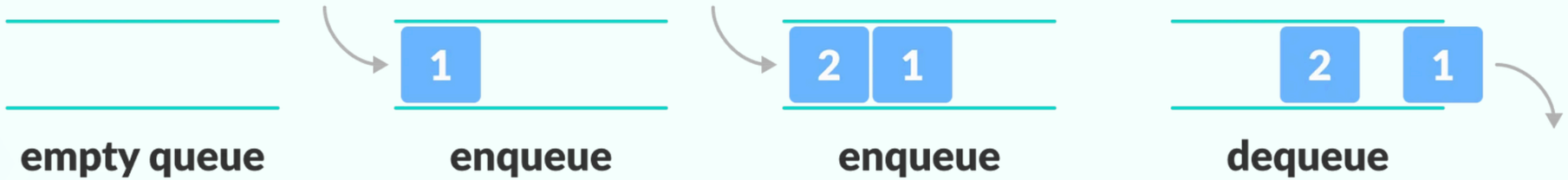
Queue - FIFO

A **queue** is a data structure that can be represented using the example of a line of people waiting in a shop at checkout.



Queue - FIFO

A queue can be defined as an ordered list which enables insert operations to be performed at one end called REAR and delete operations to be performed at another end called FRONT.



Standard Queue Operations

Enqueue

Add an item to rear.

Front

Get the front item from queue.

Dequeue

Remove an item from front.

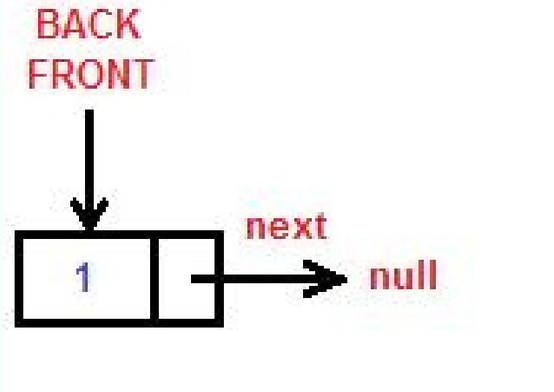
Rear

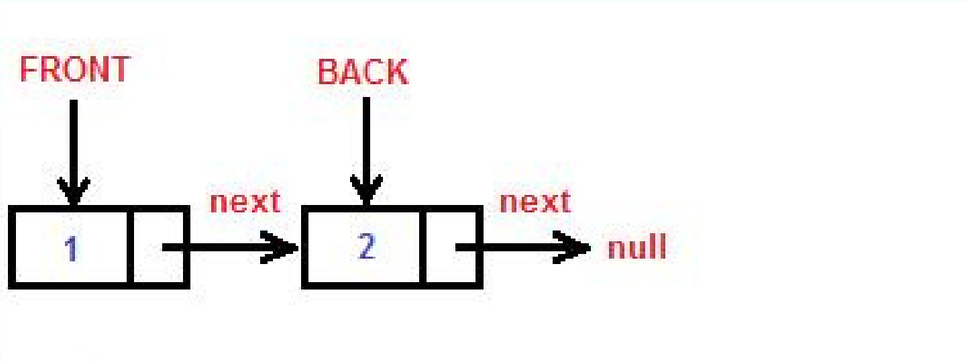
Get the last item from queue.

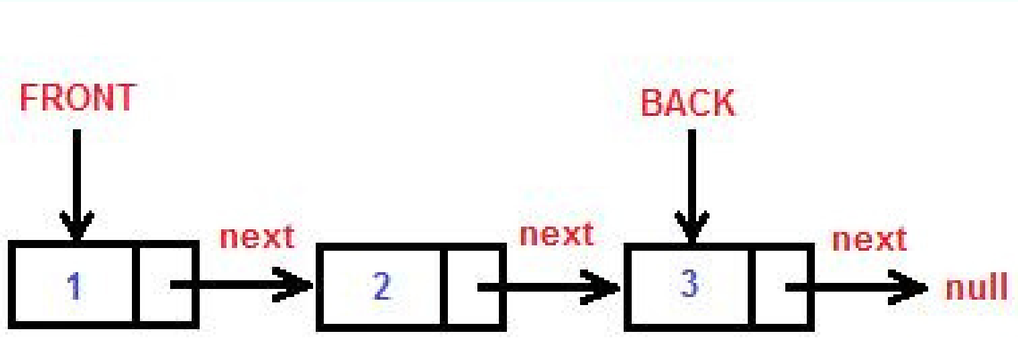
Implementation queue with linked list

Enqueue( )

- Enqueue on an empty queue
- Enqueue on a non-empty queue



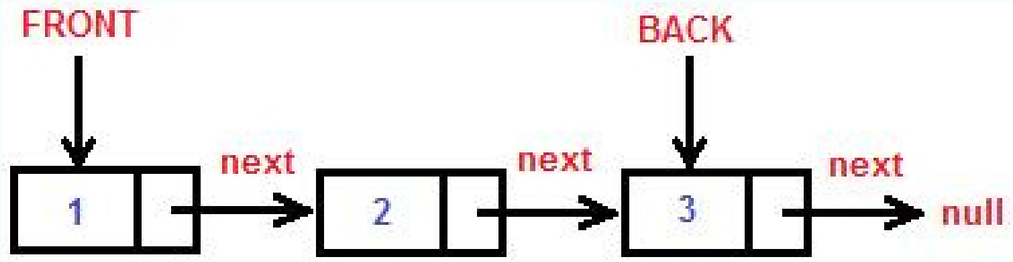


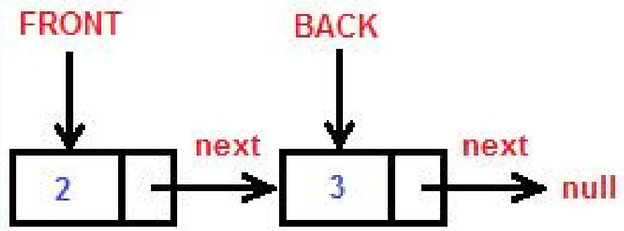


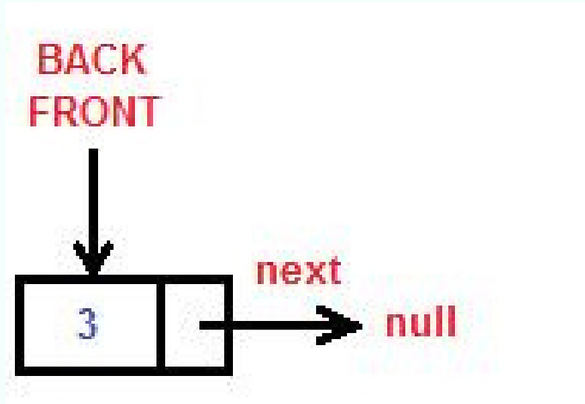
Implementation queue with linked list

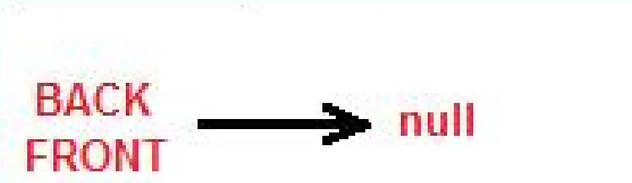
Dequeue( )

- Dequeue on an empty queue. --> None
- Dequeue on a non-empty queue. --> value



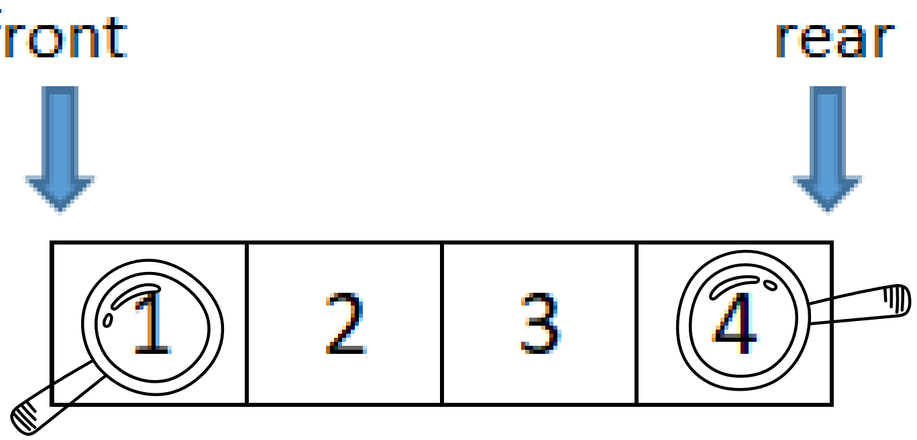






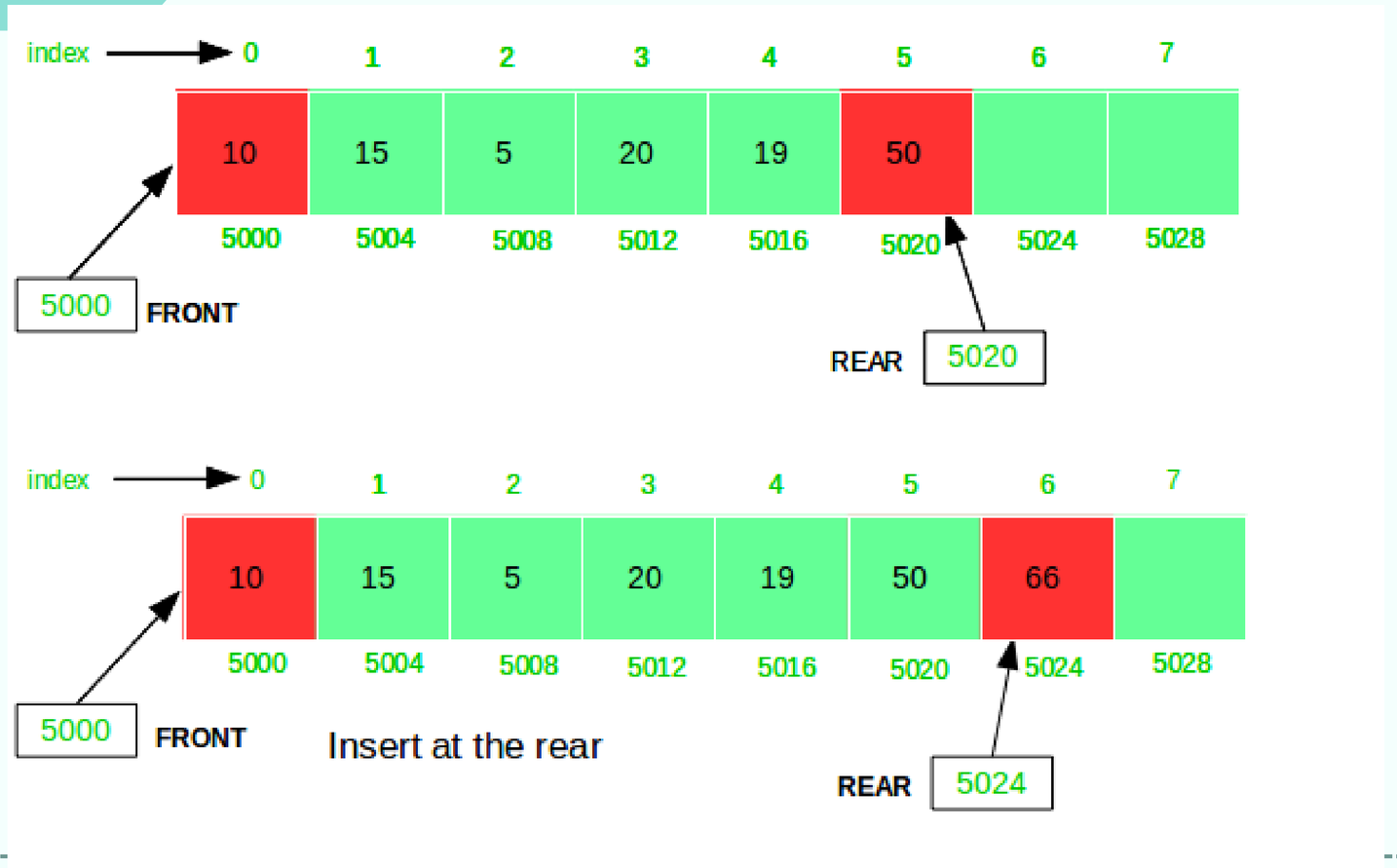
Implementation queue with linked list

Front( ) & Rear( )



Implementation queue using array

Enqueue( )

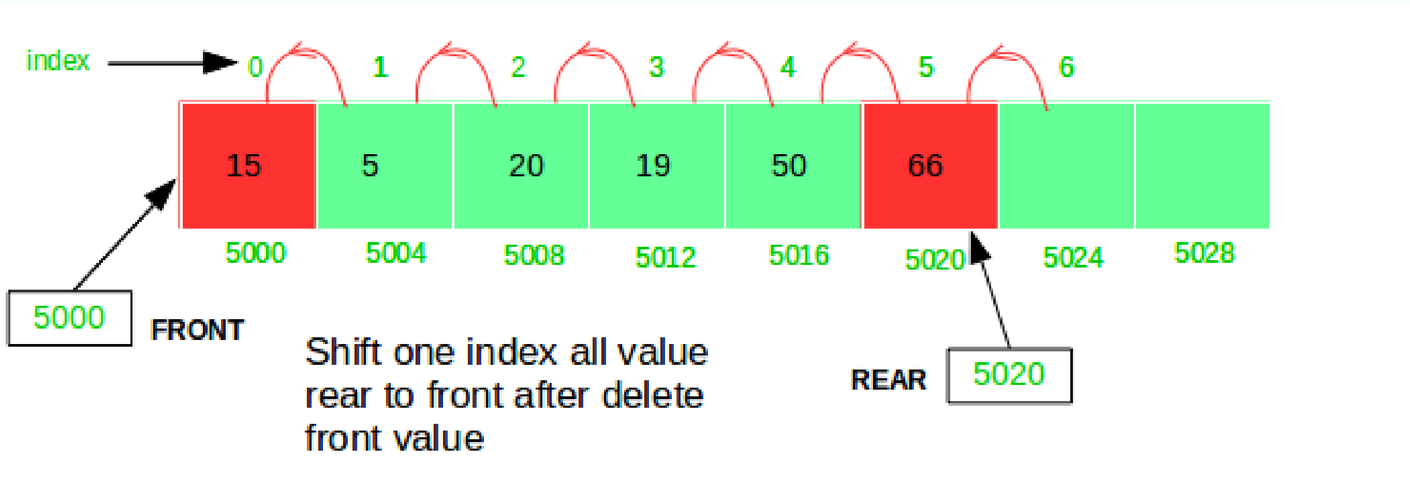


Implementation queue with Array

- create an array of size N.
- take two variables front and rear both of which will be initialized to -1 which means the queue is currently empty.
- Element
  - rear is the index of the last element added to the linked list.
  - front is the index of the first element of the array.

Implementation queue using array

Dequeue( )



Applications which solved by stack & queue

Stack

- Function calls and recursion.
- Memory Stack.
- Undo/Redo operations.
- Browser history.

Queue

- Task Scheduling.
- Printer queues.
- Web servers incoming requests.
- Buffer for devices like keyboard.
- Applied to add a song at the end or to play from the front.

Question and Answer

Thank You