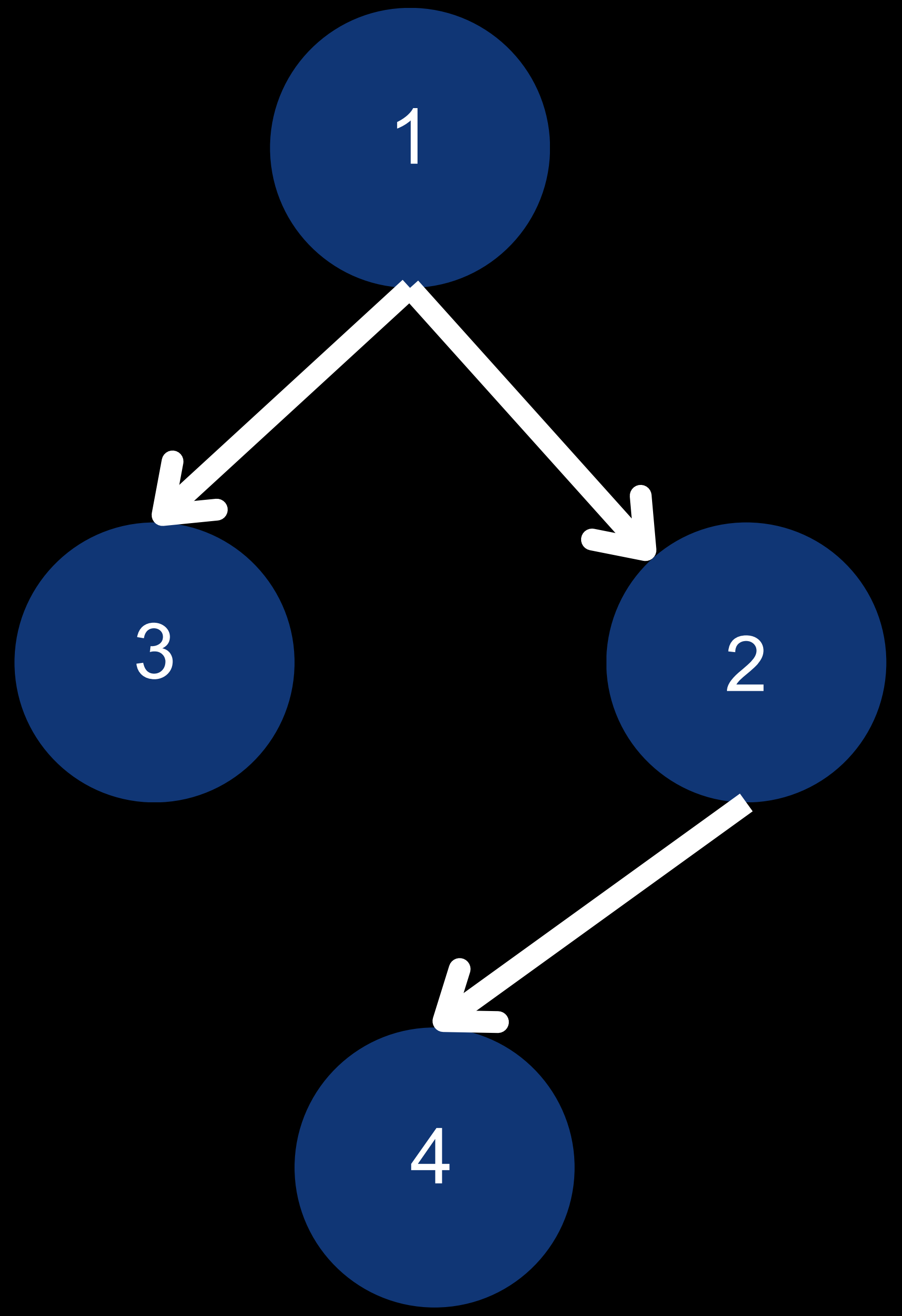
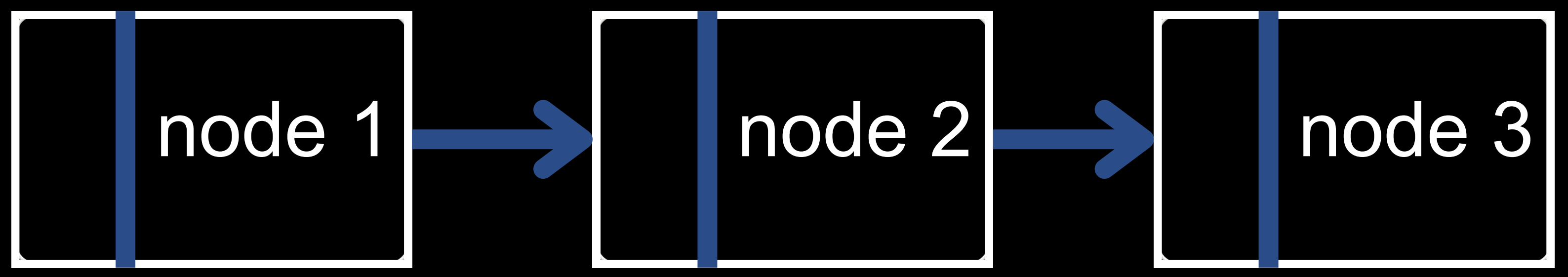
0x1A2B3C

0x4D5E6F

0x7F8A9B

DATA STRUCTURE





node 1

node 2

node 3

essammohamedst1@gmail.com

Stacks

Stack is a linear data structure based on LIFO principle

what is LIFO principle : is standard for last in last out that mean when we want to make any oeperation we make it in the first element (the top element)

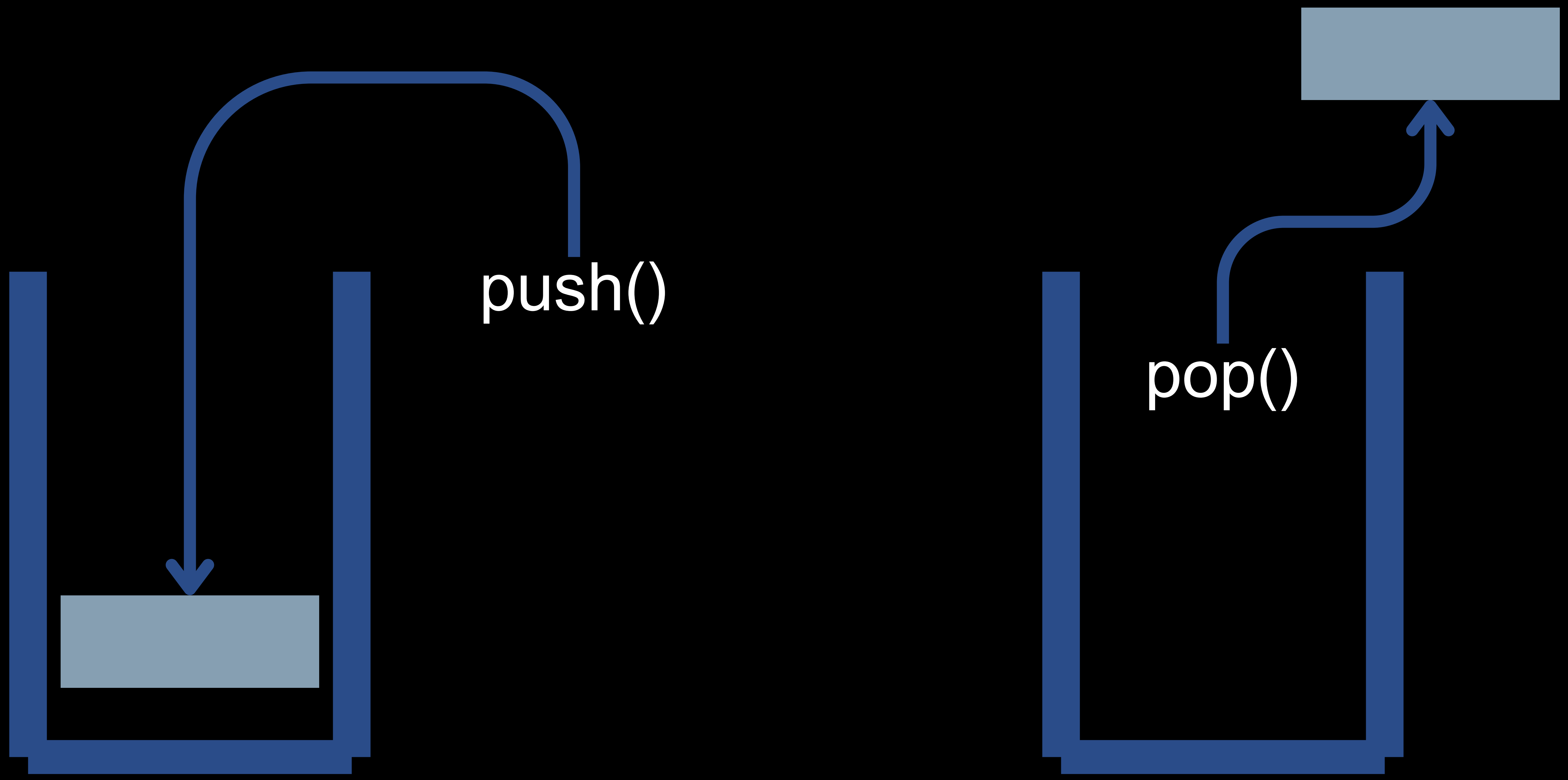
we have two types of stacks :

fixed size stack : stack have a limite size we chose it

dynamic size stack : a stack can grow and shrink as we add elemnts stack operations :

- push() to insert an element into the stack
- pop() to remove an element from the stack
- top() Returns the top element of the stack.
- isEmpty() returns true if stack is empty else false.
- isFull() returns true if the stack is full else false.

note : all operations based on LIFO principle



push()

pop()

Queues

Queue Data Structure is a linear data structure that follows FIFO (First In First Out) Principle

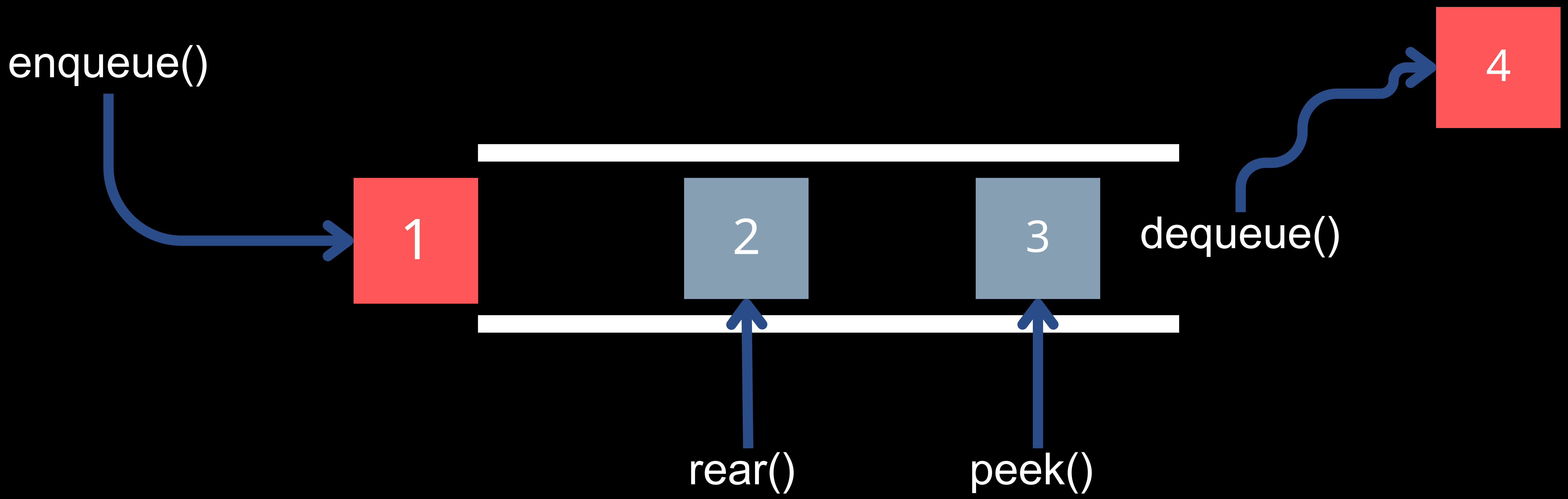
that mean the first element inserted is the first to be popped out

we have two types of queue :

- regular queue
- Dqueue is a generalized version of Queue data structure that allows insert and delete at both ends.

operations for Queue

- enqueue() – Insertion of elements to the queue.
- dequeue() – Removal of elements from the queue.
- peek() or front()- Acquires the data element available at the front node of the queue without deleting it.
- rear() – This operation returns the element at the rear end without removing it.
- isFull() – Validates if the queue is full.
- isEmpty() – Checks if the queue is empty.
- size(): This operation returns the size of the queue i.e. the total number of elements it contains.


enqueue()
4
1
2
3
dequeue()
rear()
peek()

Thanks

All code and examples can be found on GitHub https://github.com/ESSAMMOHAMED1