* Theory

* double-ended queue
A double-ended queue is an abstract data type similar to an simple queue it allow you to insert and delete from bothe means item can be added or deleted from the front and rear end.

* Algorithm
1) Add to front (Insert at front)
step1) start
step2) Check if the deque is full
step3) If the deque is empty .set both front and rear
step4) Decrement the front index by 1
step5) place the new element at the front index
step6) The element has been added.
step7) end

2) Add to rear
step1) start
step2) Check if the deque is full
step3) If the deque is empty set both front and rear to 0
step4) place the new element at the rear index
step5) Increment the rear index by 1
step6) The element has been added
step7) end

12/11/24
7:56 PM

3) Remove from front
step 1) start
step 2) Check if the dequeue is empty
step 3) store the element at the front index to return
step 4) Check if the front and rear? index are same
step 5) If the deque is not empty increment the front index by 1
step 6) Return the stored element
step 7) stop

4) Remove from Rear
step 1) start
step 2) check if deque is empty
step 3) stored the element in the rear index to return later
step 4) check if the front and rear indexes are same
step 5) If the deque is not empty decrement the rear index by 1
step 6) Return the stored element
step 7) stop

5) Get front element
step 1) start
step 2) Check if the deque is empty
step 3) Return the element at the front ind
step 4) stop

12/11/24
7:56 PM

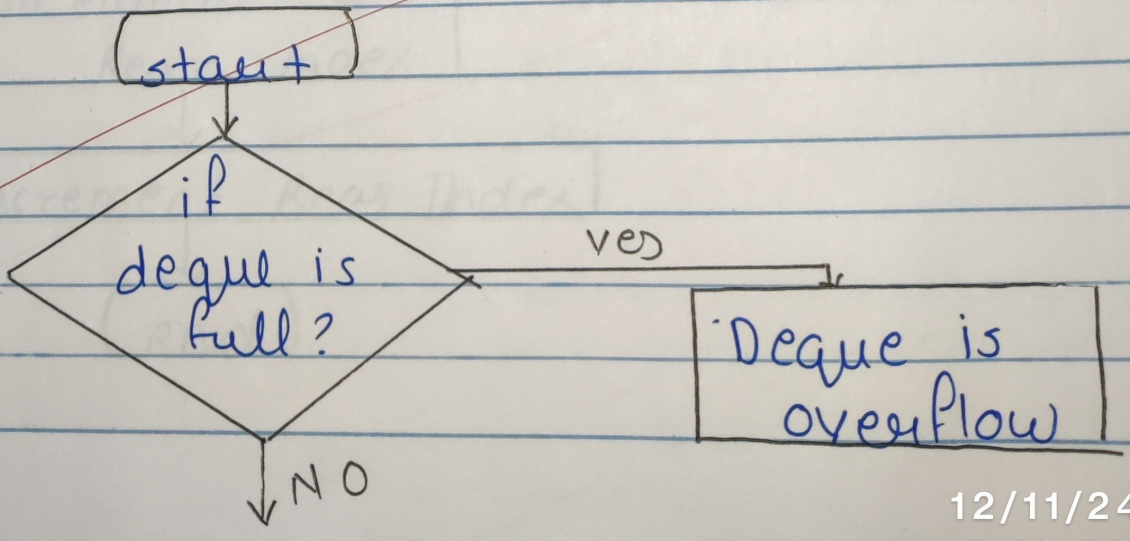
6) Get Rear element
step 1) start
step 2) Check if deque is empty
step 3) Return the element at the rear index.
step 4) end

7) Is empty operation
step 1) start
step 2) Check if the front index is -1
step 3) If it is return true
step 4) otherwise. return false
step 5) end

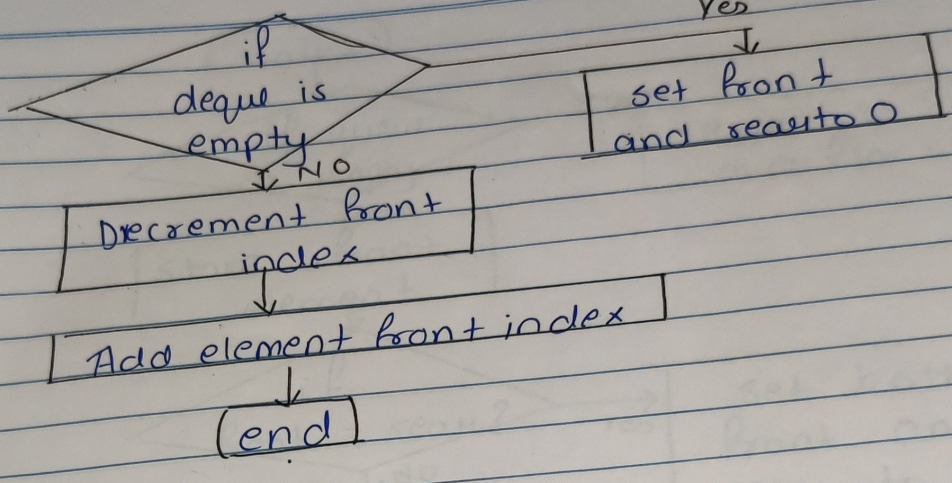
8) Is full operation
step 1) start
step 2) Check if the rear index has reached maximum size of the deque minus one
step 3) If it i' return true otherwise return false
step 4) . end.

* flowchart

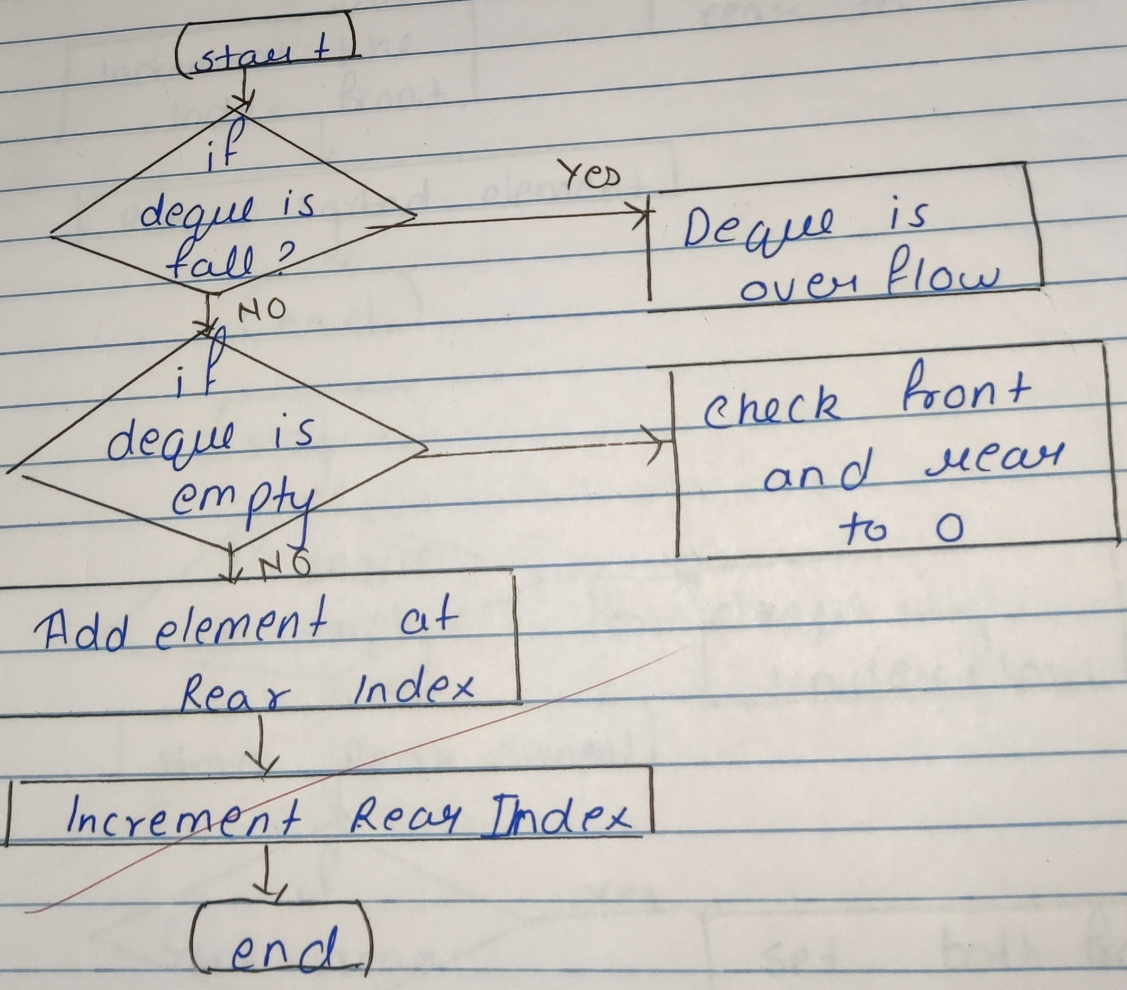
1)



12/11/24
7:56 PM

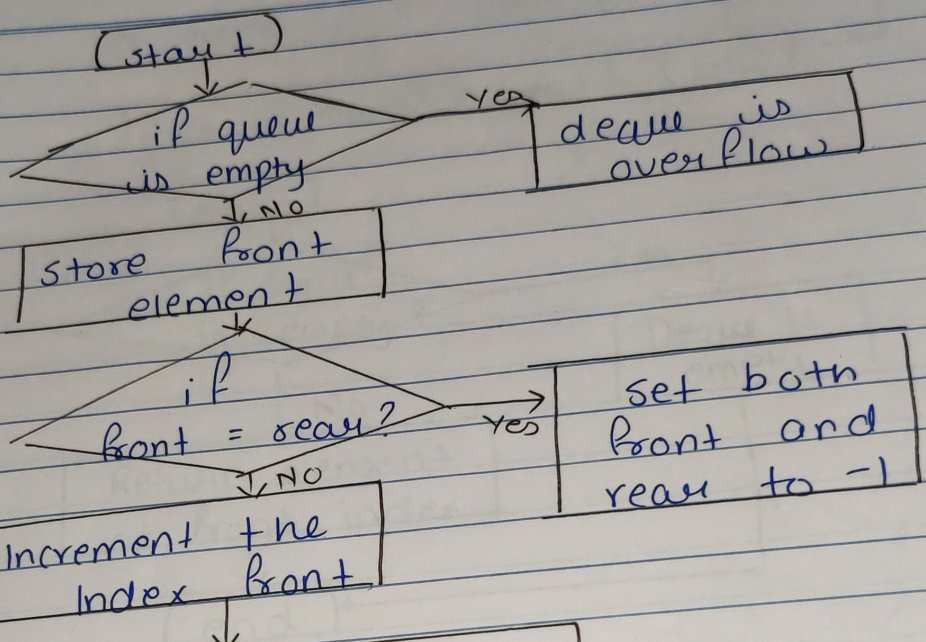


2)

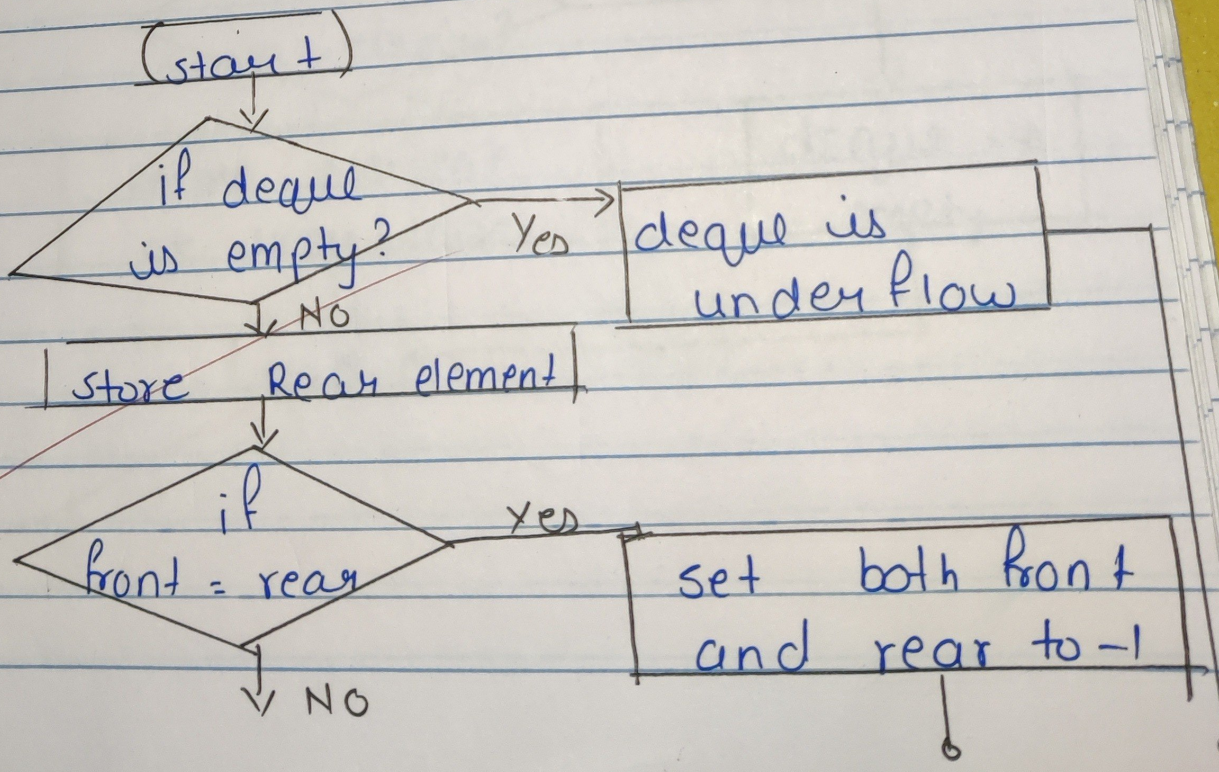


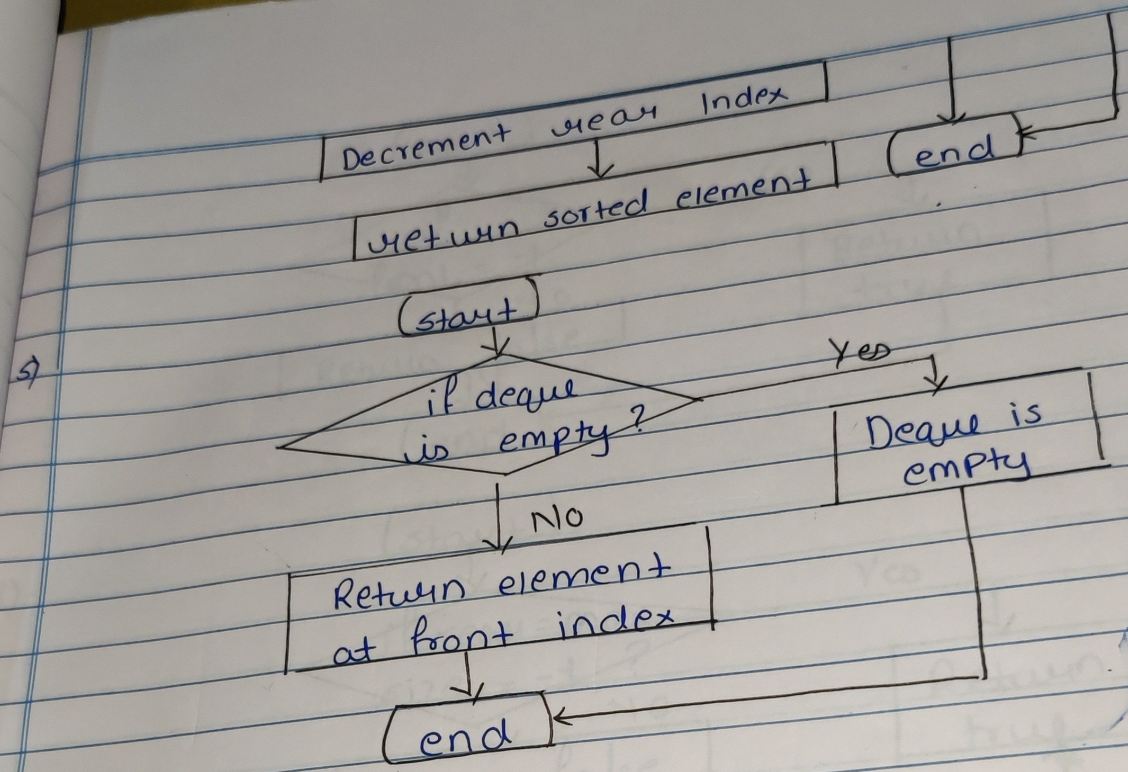
12/11/24
7:56 PM

9)


start
if queue is empty
Yes
deque is overflow
NO
store front element
if front = rear?
Yes
set both front and rear to -1
NO
Increment the Index front
Return sorted element
end.

a)

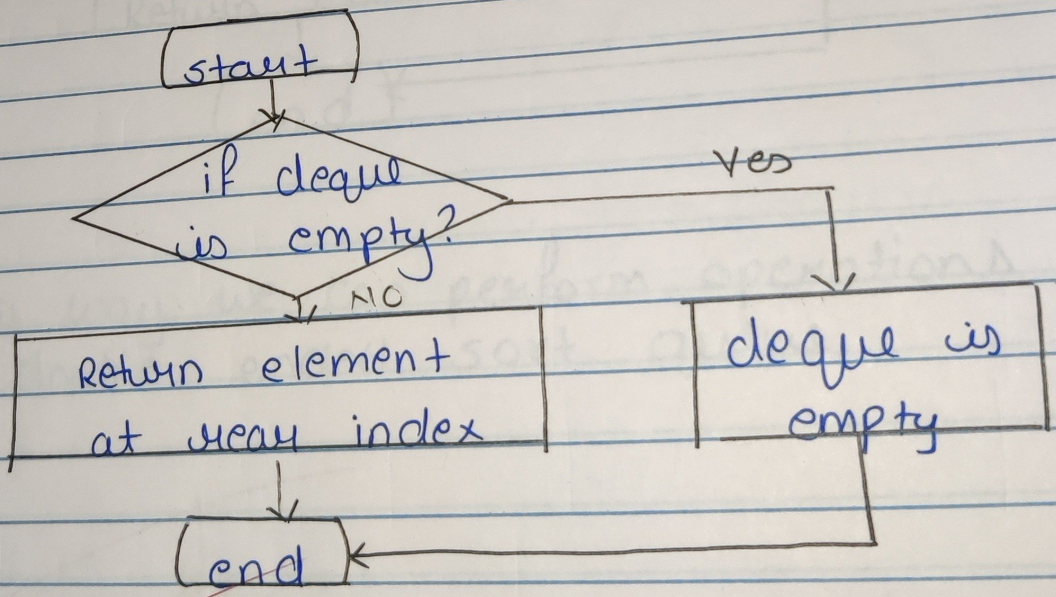

start
if deque is empty?
Yes
deque is underflow
NO
store Rear element
if front = rear
Yes
set both front and rear to -1
NO

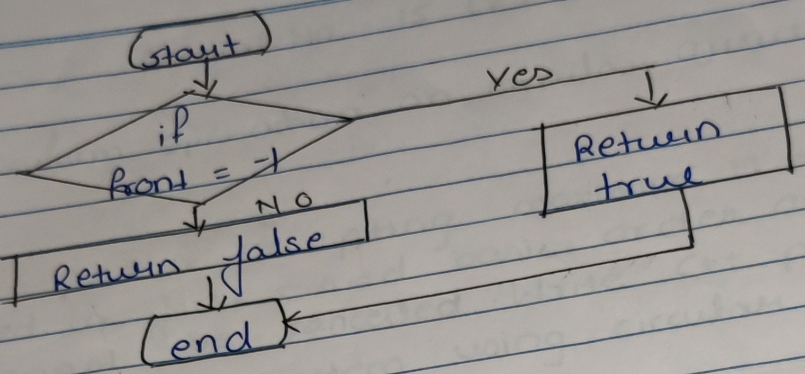

Decrement rear index
return sorted element
end

5)

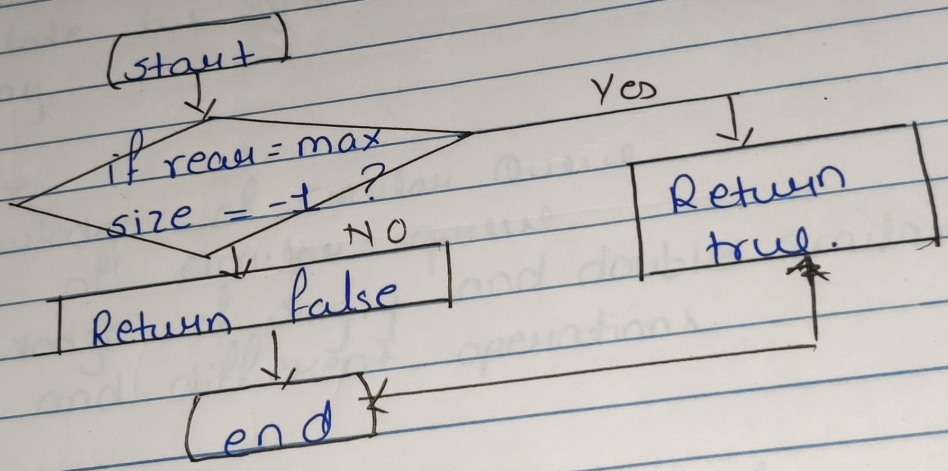
start
if deque is empty?
Yes
Deque is empty
No
Return element at front index
end

6)


start
if deque is empty?
Yes
No
Return element at rear index
deque is empty
end



8)



* Conclusion

By this way we can perform operations on the double ended sort queue.

12/11/24
7:56 PM