Computer Science & IT

Data Structure & programming

Queue

Lecture No. 02

By- Abhishek Sir

Recap of Previous Lecture

- Topic — Queue
- Topic — Enqueue & dequeue
- Topic
- Topic
- Topic

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

Circular Queue

How to implement using stack

Slide

Topic : Question

#Q.

Compute the post fix equivalent of the following

expression 3 × log (x + 1) – a /2

= 3 * log x1+ – a/2

= 3 * x1+log – a/2

= 3x1+log* – a2/

= 3x1+log* a2/ –

Topic : Question

A function f defined on stacks of integers satisfies the following properties. $f(\phi) = 0$ and $f(push(S,i)) = \max(f(s), 0) + 1$ for all stacks S and integers i. If a stack S contains the integers 2, −3, 2, −1, 2 in order from bottom to top, what is f(S)?

(A) 6 (B) 4

(C) 3 (D) 2

$f(\boxed{2|-3|2|-1|2}) = f(push_S(\boxed{2|-3|2|-1}, 2))$

$= \max(f(s), 0) + 2$ (1)

$f(\boxed{2|-3|2|-1}) = f(push_S(\boxed{2|-3|2}, -1))$

$= \max(f(\boxed{2|-3|2}), 0) - 1$ (2)

$f(\boxed{2|-3|2}) = f(push_S(\boxed{2|-3}, 2))$

$= \max(f(\boxed{2|-3}), 0) + 2$ (−1)

$f(\boxed{2|-3}) = f(push(\boxed{2}, -3)) = \max(f(\boxed{2}), 0) - 3$ (2)

$$f(\boxed{12}) = f(\text{push}(\square, 2))$$

$$= \max(f(\phi), 0) + 2$$

$$0 + 2 = \textcircled{2}$$

Topic : Circular Queue

```
void Enqueue(int data){
    if((rear==Max-1 && front==0) || (front==rear+1)){
        printf("queue is full");
        return;
    }
    rear=(rear+1) % Max;
    a[rear]=data;
    if(front==-1)
        front=0;
}
```

front = 2
rear = 1

front = (rear+1) % Max

front = 0 rear = 4

Single condition

Slide

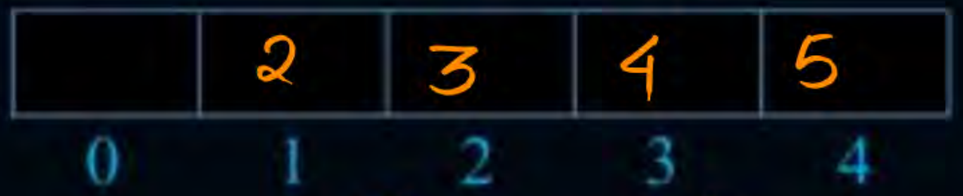
Topic : Circular Queue

```
int dequeue( ) {
    int data;
    if (front == -1){ // if (IsEmpty())
        printf ("queue is empty");
        return -1;
    }
    data = a[front];
    if (rear == front)
        rear = front = -1
    else
        front = (front+1)% Max
    return data;
    }
```

Slide

Topic: Circular Queue

front = 1, rear = 4



Enqueue(6)

rear = Max - 1

overflow

Remainder

rear = (rear + 1) % Max

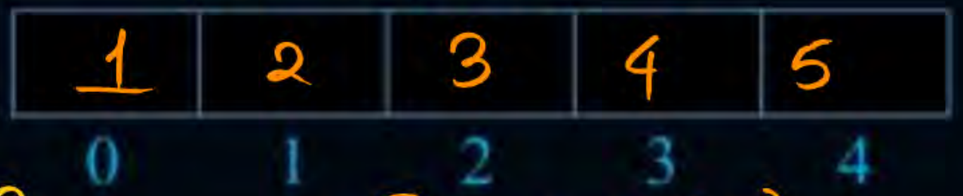
= (4+1) % 5

= 5 % 5 = 0

Slide

Topic: Circular Queue

front = 0, rear = 4

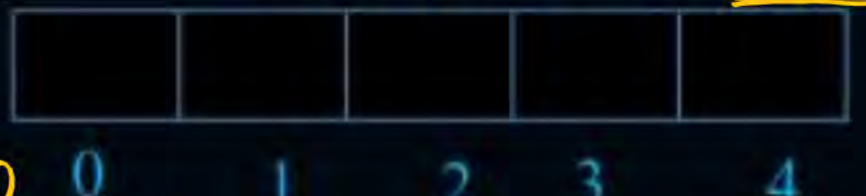


front = 1
rear = 4

Enqueue (6)

Enqueue (6)

rear = Max - 1
overflow && front = 0 ✓



↑ first full condition

front = 1
rear = 0

↑ Is queue is full ?

① rear < front

② front = rear + 1 ✓

Slide

Topic: Circular Queue

rear = 0
front = 1

Dequeue()

rear = 0
front = 2

Enqueue(7)

rear = 0
front = 2

rear = 1
front = 2

rear = (rear + 1) % Max

Slide

Topic: Circular Queue

front = 2
rear = 1

Enqueue(8)

front = 2, rear = 1

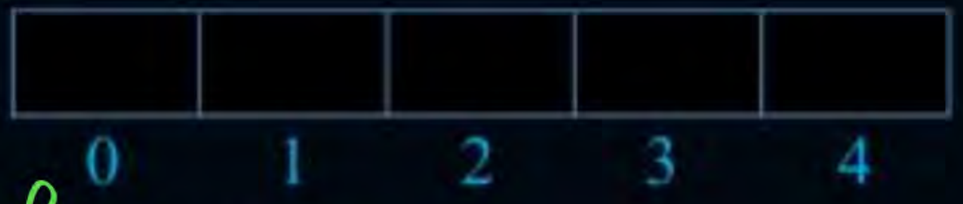
overflow

front = rear + 1

rear < front

Slide

Topic: Circular Queue



front = 4, rear = 1

front = 4 , rear = 2

Enqueue(8)

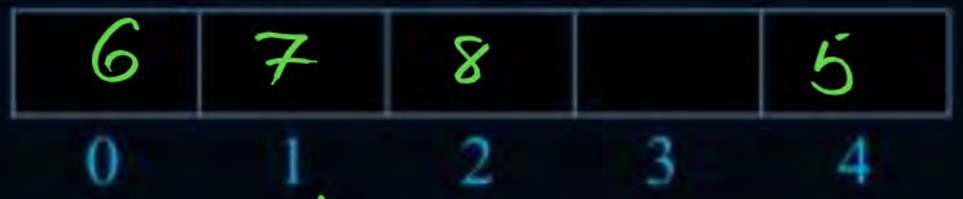
Condition = front == (rear+1) % Max

(4 == 2)

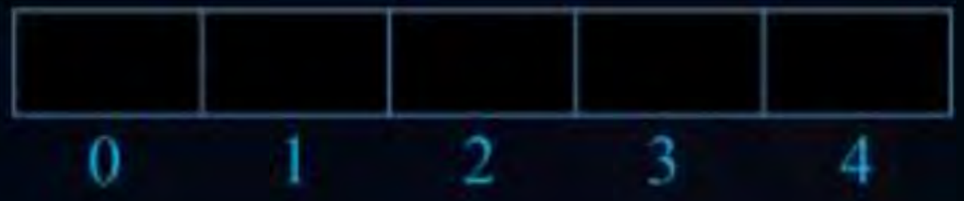
Slide

Topic: Circular Queue

front = 4, rear = 2



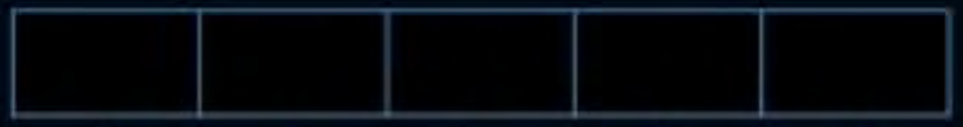
dequeue()

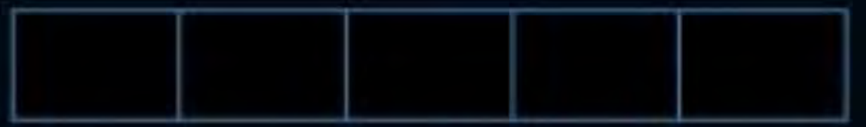


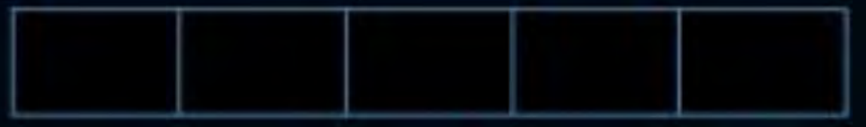
front = 0, rear = 2

Slide

Topic: Circular Queue







Topic : Circular Queue

If following operation are performed on queue of size 5.

Enqueue(11), Enqueue(12), Enqueue(13)

dequeue(), dequeue(), Enqueue(16), Enqueue(20)

Enqueue(29) Enqueue(70), X = dequeue()

if y is Index of front & z is Index of rear then X+Y-Z is 15

Slide

11), Enqueue(12), Enqueue(13)

), dequeue(), Enqueue(16), Enque(20

(29) Enqueue(70), X = dequeue()

Index of front & z is Index of rear then

(15)

rear=2

11	12	13	
0	1	2	3

29	70	13	16
0	1	2	3

front = ~~2~~ 3

rear = 1

X

13

Implementation of Queue Using Stack

* Single stack can't implement using push, pop
* Auxiliary storage. another stack is used

Slide

Implementation of Queue Using Stack

1. Suppose S_1, S_2 are two Stacks

Algorithm — Enqueue = push(S_1)

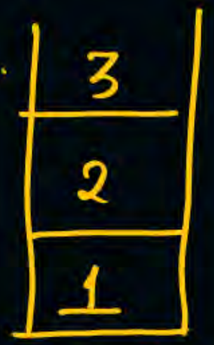
dequeue() = 1 Empty S_1 by poping element from S_1 and push S_2

2 pop element from S_2 and return it

3. Restore content of S_1 by poping element from S_2 & pushing it in S_1 ✓

Slide

1.



S_1

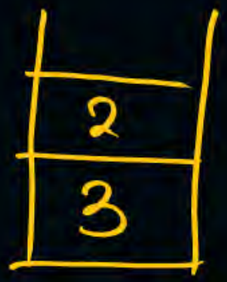
Enqueue(1) = push(1)

Enqueue(2) = push(2)

Enqueue(3) = push(3)

2.

S_1



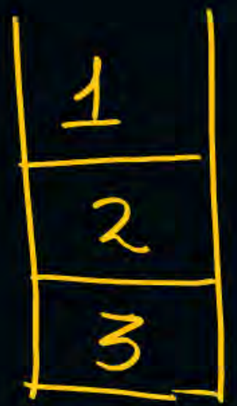
S_2

return 1

dequeue()

1.

S_1



S_2

3.

S_1

S_2

THANK - YOU

Slide