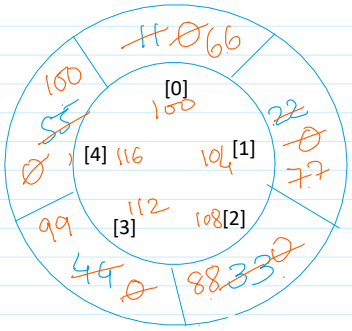
Circular Queue

28 October 2023 17:29

= -1

front

rear



front = 1 rear = 0
front = 2 rear = 1
front = 3 rear = 2
front = 4 rear = 3

front == (rear+1)

0 == (4+1)

front == (rear+1) % SIZE

0 == (4+1) % 5

0 == 0

Enqueue:

1) Check if queue is not full
2) Increment rear as
   Rear = (rear+1) + SIZE
3) Add element at rear position
4) If front == -1, make front = 0

Dequeue :

Check if Queue is not empty.
Increment front .
If front is 4 and rear is 0
to delete the rear position, we cannot increment front as front ++ Will be index 5 but we want to delete index 0;
So,
Front = front +1 %SIZE

If deleting the last element in queue
If(front == rear)
Front = rear = -1

rear = rear +1

rear = (rear+1) % SIZE

rear = (-1+1) % 5 → 0 % 5 = 0

(0+1) % 5 → 1 % 5 → 1

(1+1) % 5 → 2 % 5 → 2

(2+1) % 5 → 3 % 5 → 3

(3+1) % 5 → 4 % 5 → 4

rear = (4+1) % 5 → 5 % 5 → 0

Queue Empty condition
If(rear == -1) queue is empty

Queue Full condition :
When Queue is full,

Rear = 0 front 1
Rear = 1, front = 2
Rear = 2 front = 3
Rear = 3 front = 4
rear = 4, front = 0

This means,
Front == rear + 1

But when rear = 4 front =0
0 == 4+1 does not satisfy the above condition
So the queue full condition can be
Front == (rear+1) %SIZE
0 == (4+1) % 5
0 == 5%5
0==0

front++

front = front +1

front = (front +1) % SIZE

(1+1) % 5 → 2 % 5 → 2

= (2+1) % 5 → 3 % 5 → 3

= (3+1) % 5 → 4 % 5 → 4

front = (4+1) % 5 → 5 % 5 → 0

Option 2:

```
If rear == SIZE-1
    Rear = 0;
Else
    Rear++;
```