Queue

FIFO

=

first In & first out

R

rear

front

- Movie ticket line
- job queues
- printer

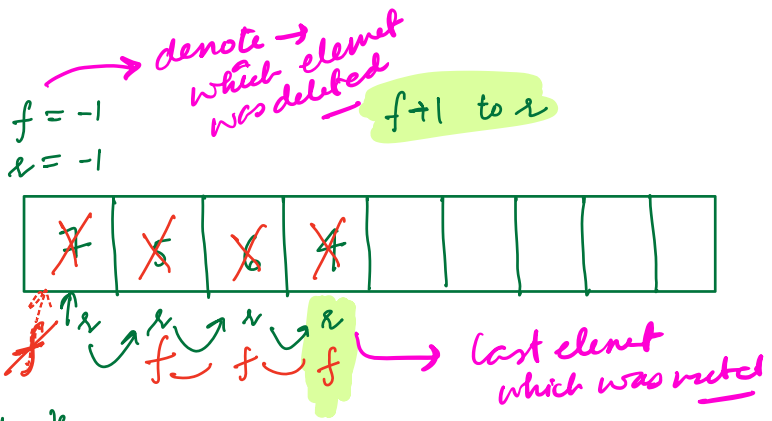
operations by queue :-

- Enqueue(x) ⟶ insert at an end (rear)
- Dequeue() ⟶ delete from an end (front)
- front()
- isempty()

O(1)

Implementation

- Arrays



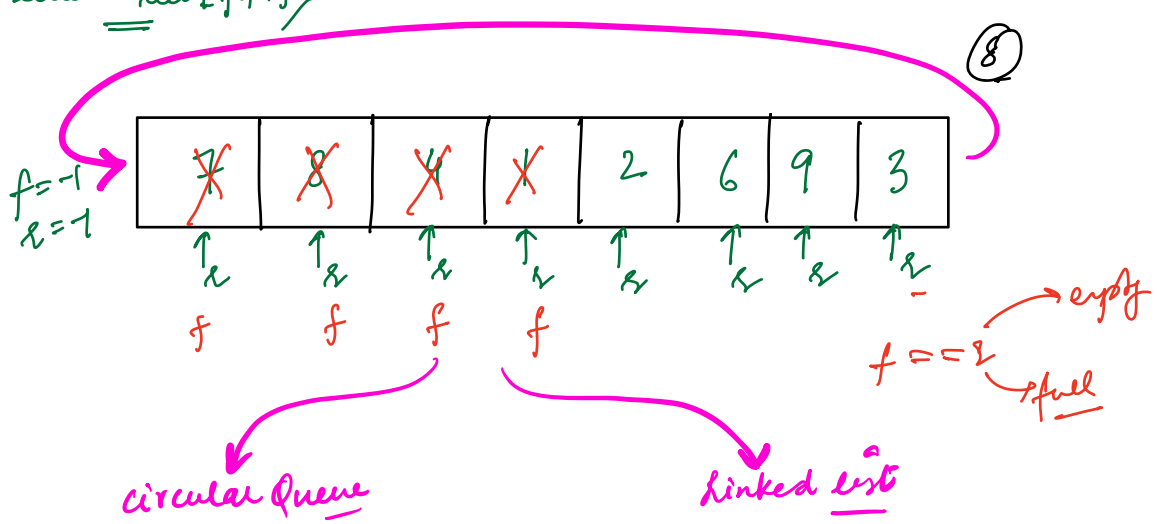
7 5 6 4 dequeue front()

```
void enqueue(x)
{  if(r == n-1){ overflow }
   else { r++;
          arr[r]=x; }
}
```

```
void dequeue()
{ if(q.isempty()){ no elmt}
  else  f++;
}
```

```
bool isempty()
{
   if(f==r)
     return true;
   return false;
}
```

```
int front()
{   if( !empty)
    return arr[f+1];
}
```



size = 0;

total elements in queue

```
void enqueue ( int x)
{   if( size == N) { overflow }
    else { r = (r+1) % n;
           arr[r] = x;
           size++;
         }
}
```

```
void dequeue ( )
{   if ( size == 0) { // empty }
    else
    {   f = (f+1) % n;
        size--;
    }
}
```

```
int front ()
{ if(size == 0) { empty }
  else
  {  return
     arr[(f+1)%N];
  }
}
```

```
bool isempty ( )
{
    if (size == 0) return true;
      else false;
}
```

- LL

enqueue → insert at tail

head , tail

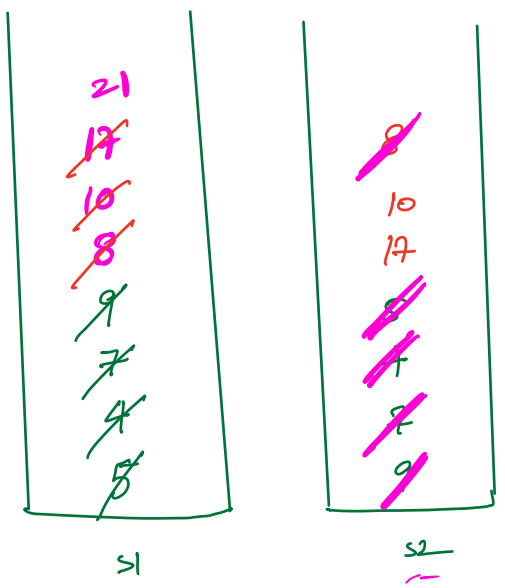
dequeue → delete from head

head = head.next;

```
front ()
{   head.data;
}
```

- Implement queues using stacks (s).

5 4 7 9 ↓ 8 10 ↓ ↓ 17 ↓ ↓ 21



We can't implement a queue using a single stack.

O(1)

```
void enqueue(int x)
{
    s1.push(x);
}
```

O(1) amortized

```
int front()
{
    if(s2.empty())
    {
        while(!s1.empty())
        {
            s2.push(s1.top());
            s1.pop();
        }
    }
    if(!s2.empty())
        return s2.top();
}
```

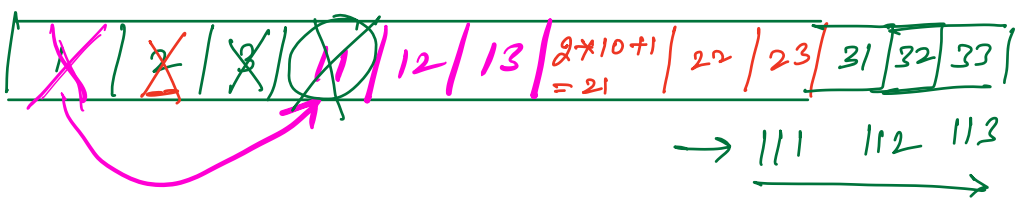
asymptotic O(n) → O(1) amortized

```
void deque()
{
    if(s2.empty())
    {
        while(!s1.empty())
        {
            s2.push(s1.top());
            s1.pop();
        }
    }
    if(!s2.empty())
        s2.pop();
}
```

Q N^{th} number → 1, 2, or 3

1	31	N^{th}
2	32	
3	33	
11	111	
12	112	
13	113	
21		
22		
23		

B.F :- start from 1 ---- , check if it's valid number count



→ 111 112 113

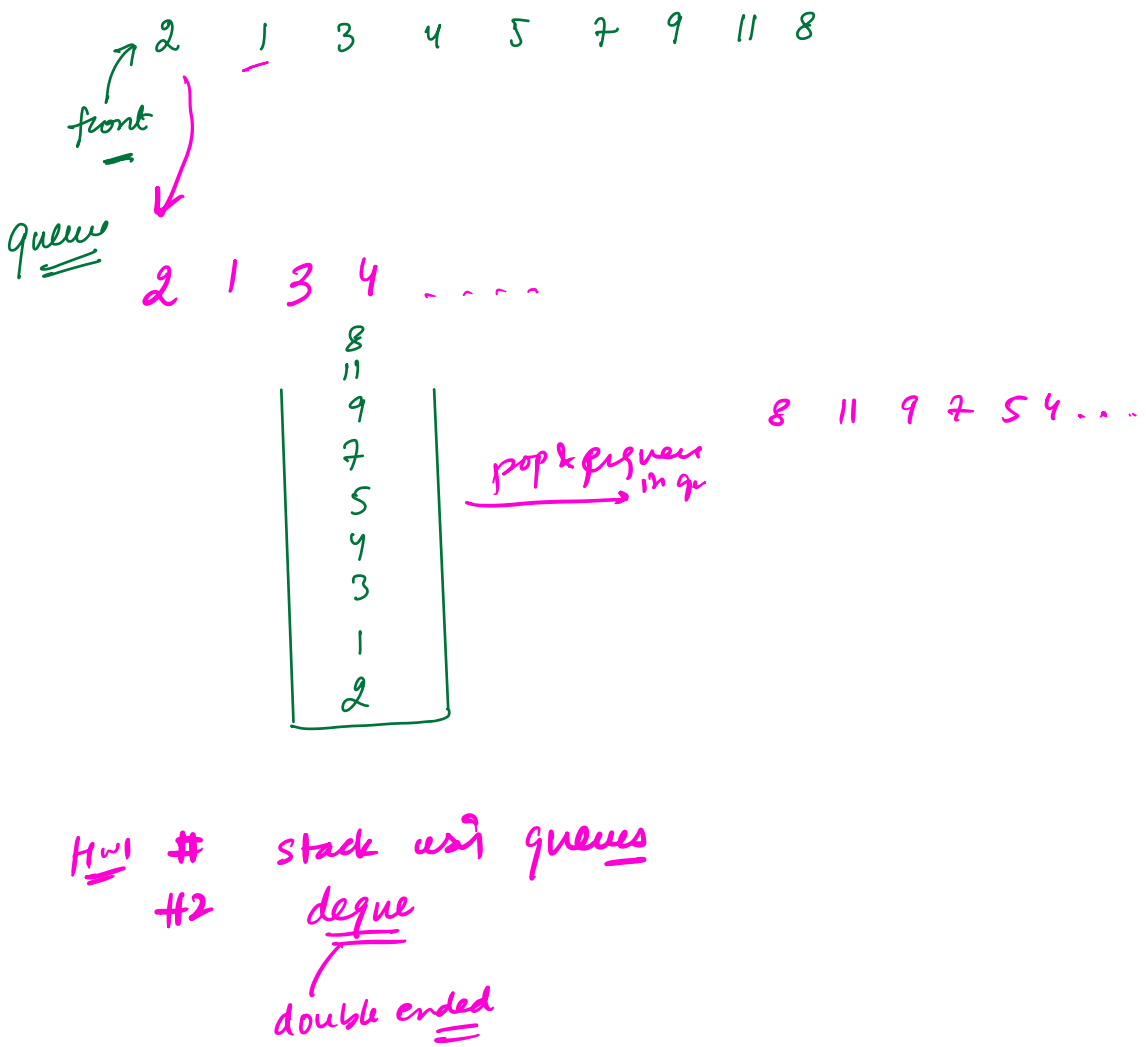
Q N^{th} perfect number.

↳ • even length
• palindrome
• 1, 2

11
22
1111
1221
2112
2222
11111111

aa'
↓ reverse

- Reverse a queue



Hw1 # stack using queues

#2 deque

double ended