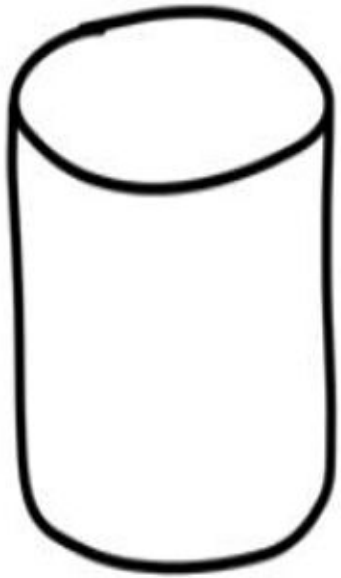
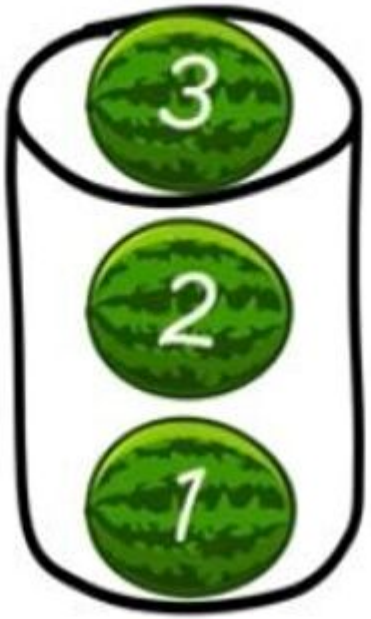
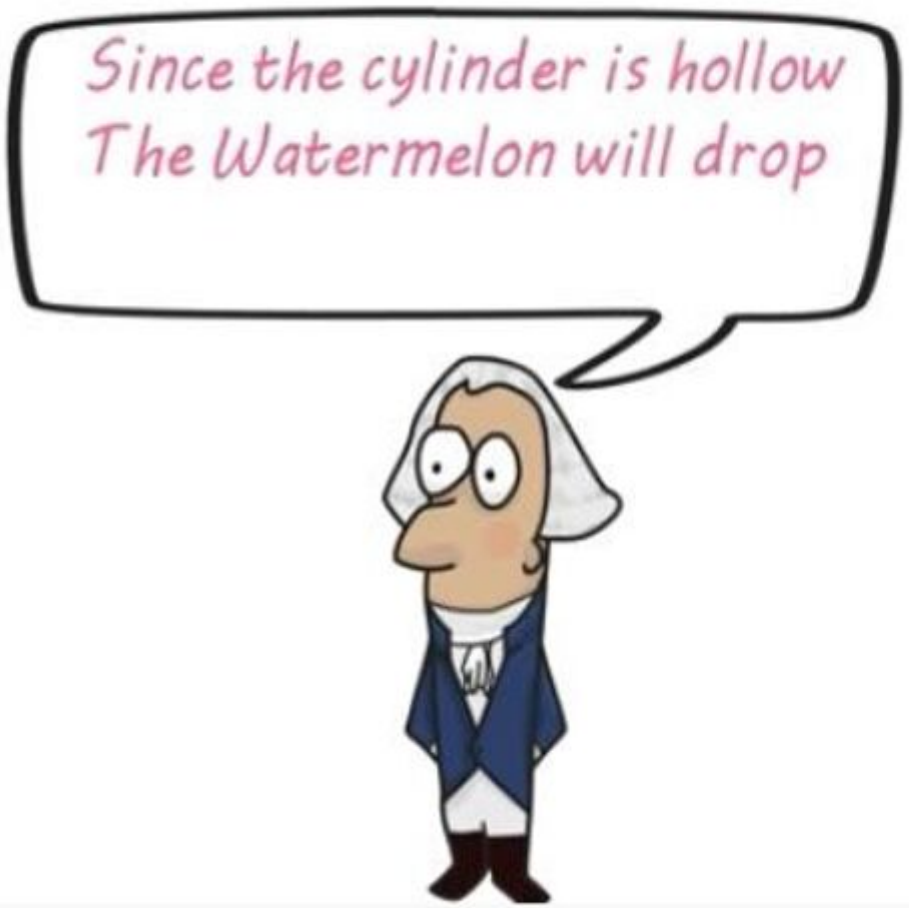
Queue is extremely easy

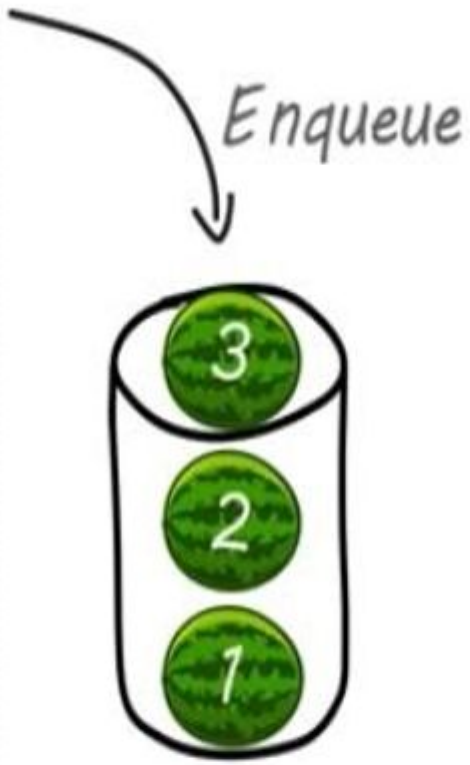


A Queue is like a empty
Hollow cylinder
Means here both the ends are
Open

Let's put these Watermelon's
in the hollow Cylinder


3
2
1

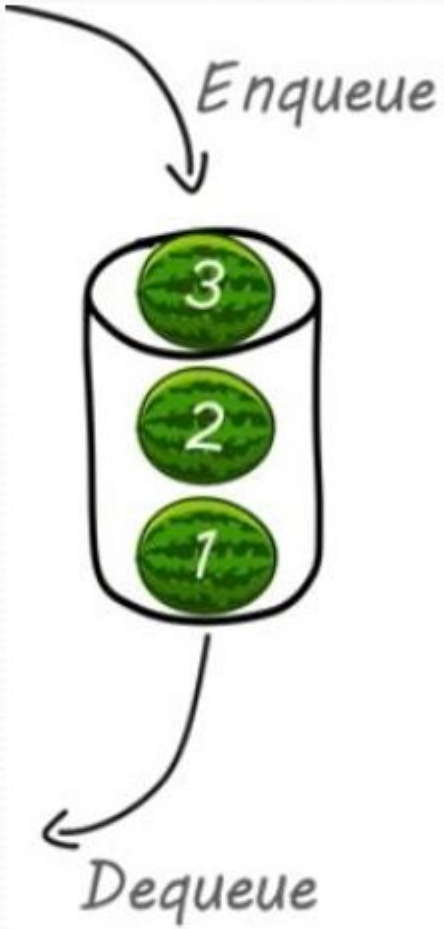

Since the cylinder is hollow
The Watermelon will drop



Operations

1. We will insert element in queue

It is called as Enqueue



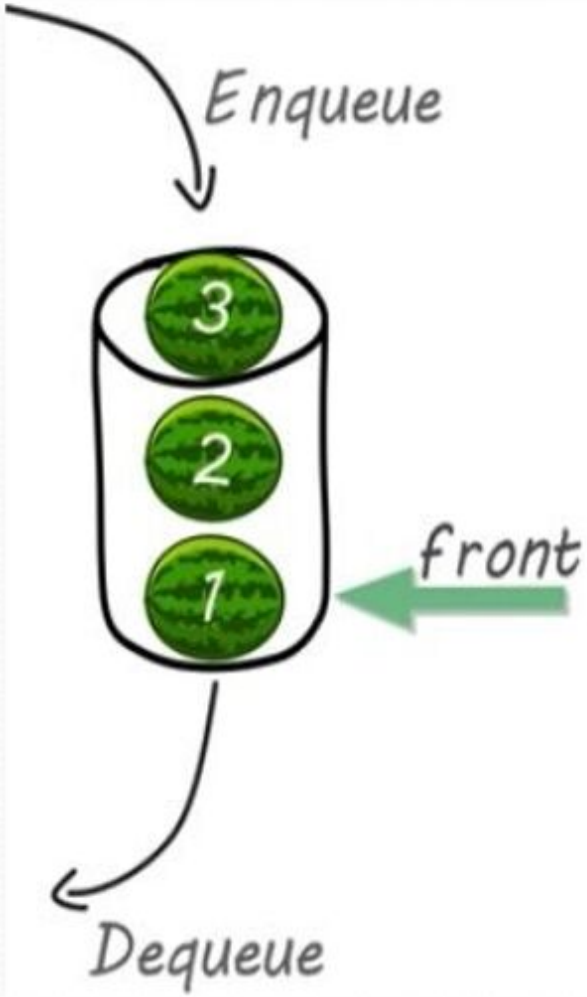
Operations

1. We will insert element in queue

It is called as Enqueue

2. Remove element from queue

It is called Dequeue



Operations

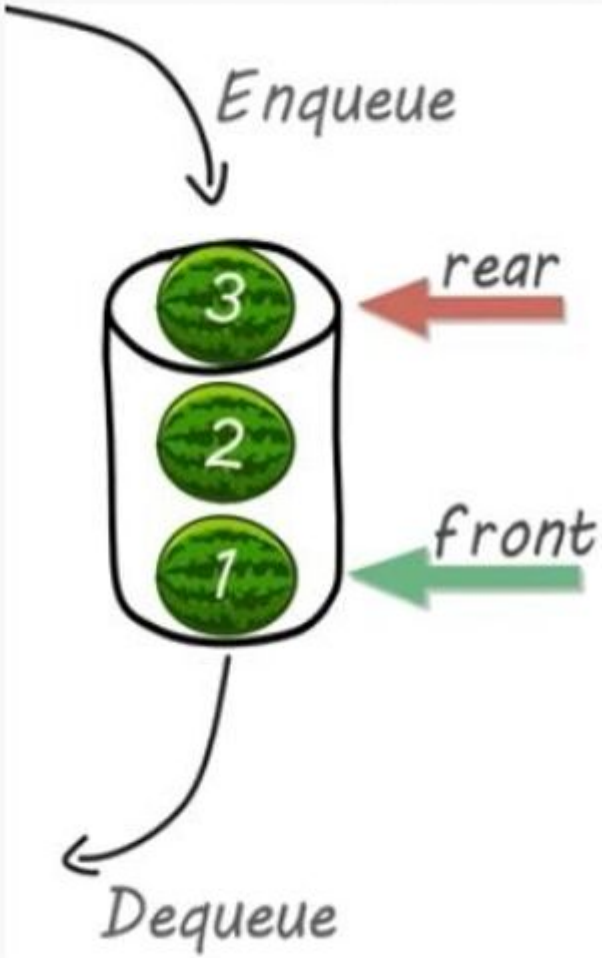
1. We will insert element in queue

It is called as Enqueue

2. Remove element from queue

It is called Dequeue

3. front , which will return the front element in queue



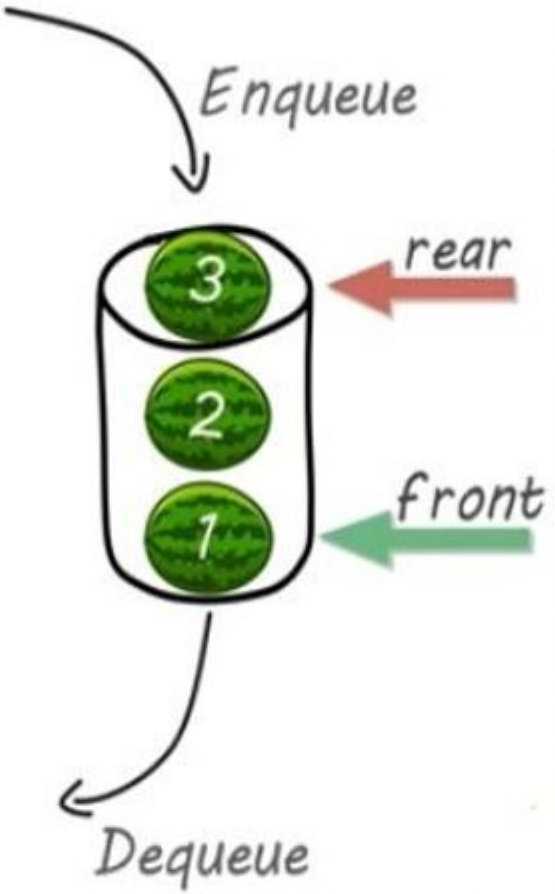
Operations

1. We will insert element in queue
   It is called as Enqueue
2. Remove element from queue
   It is called Dequeue
3. front , which will return the front element in queue
4. rear , to get the last element fro queue

In Queue , the deletion is done from front

If I remove the watermelon's they will be in the order

1 2 3

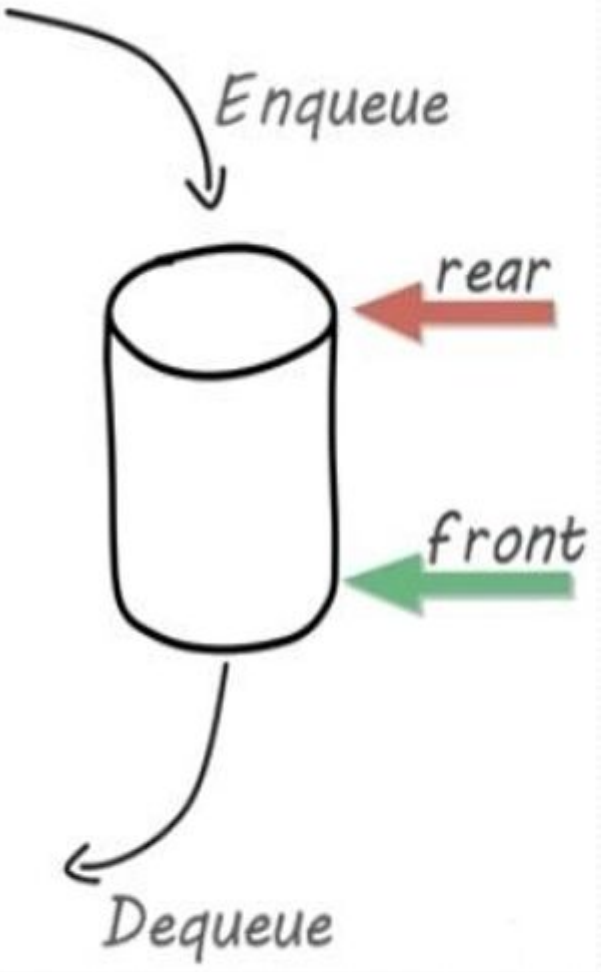


In Queue , the deletion is done from front

If I remove the watermelon's they will be in the order

1 2 3 1 2 3

If I insert the element then it will inserted from backside



Queue is called FIFO
(First In First Out)

Means the element which was inserted
first comes out first

example 1 watermelon

Queue ADT

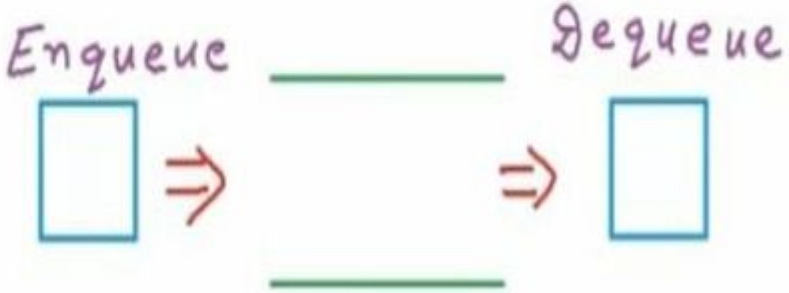
A list or collection with the restriction that insertion can be performed at one end (rear) and deletion can be performed at other end (front).

Queue ADT

Queue - First-In-First-Out
(FIFO)

Stack - Last-In-First-Out
(LIFO)

Queue ADT



Operations

(1) EnQueue(x)

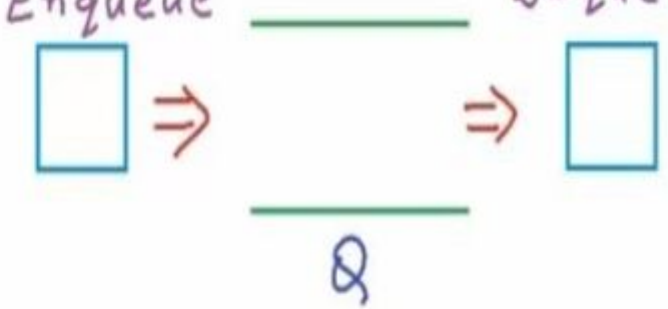
(2) Dequeue()

(3) front()

(4) IsEmpty()

} Constant time or O(1)

Queue ADT



Operations

(1) EnQueue(x)

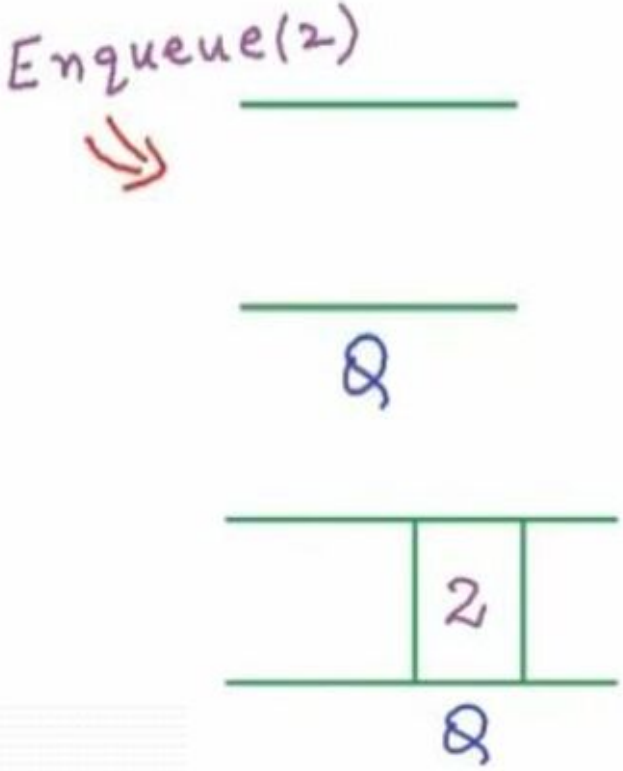
(2) Dequeue()

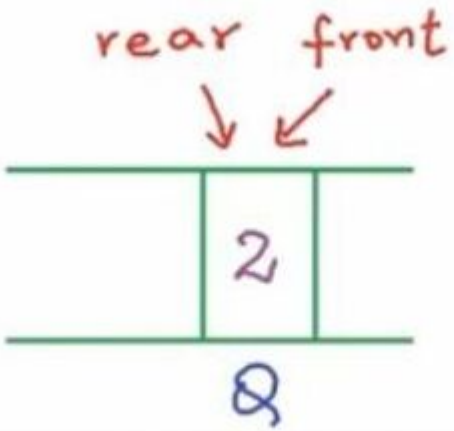
(3) front()

(4) IsEmpty()

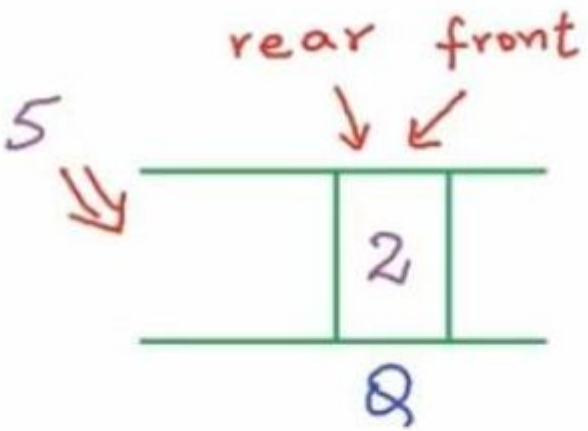
Queue ADT

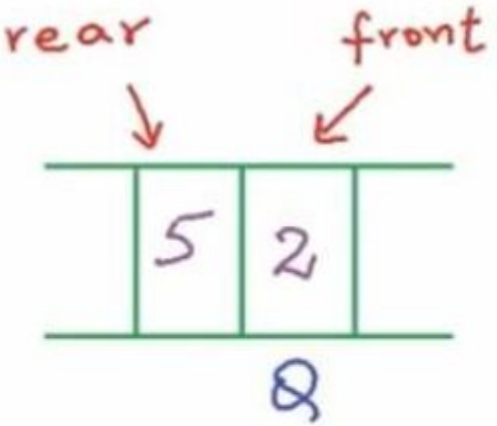
Enqueue(2)

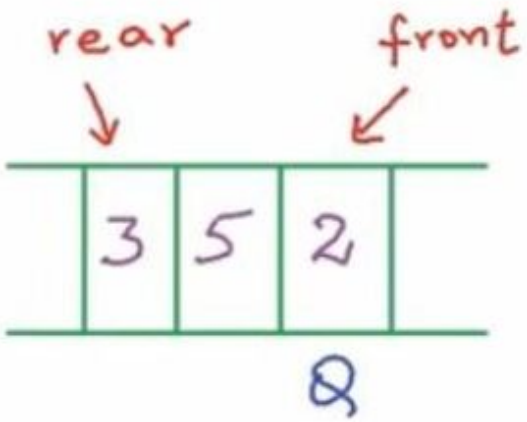




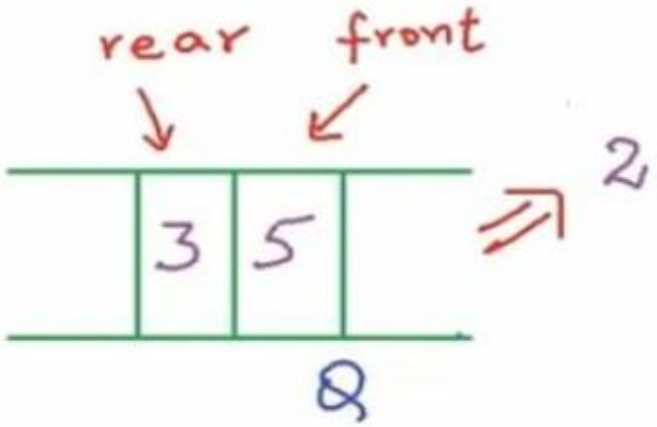
Enqueue(2)
Enqueue(5)



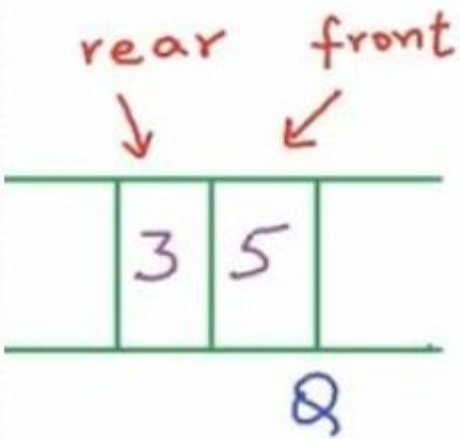




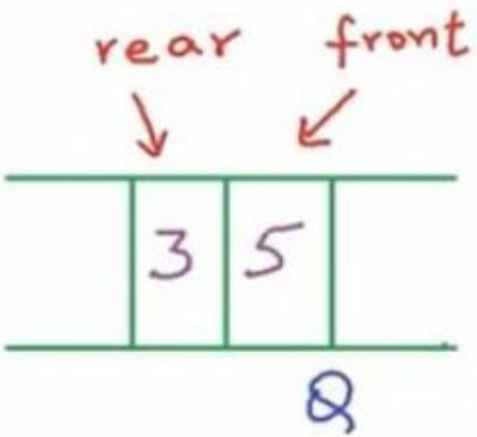
Enqueue(2)
Enqueue(5)
Enqueue(3)



Enqueue(2)
Enqueue(5)
Enqueue(3)
Dequeue()



Enqueue(2)
Enqueue(5)
Enqueue(3)
Dequeue() ⇒ 2



Enqueue(2)
Enqueue(5)
Enqueue(3)
Dequeue() ⇒ 2
front() ⇒ 5
IsEmpty() ⇒ false