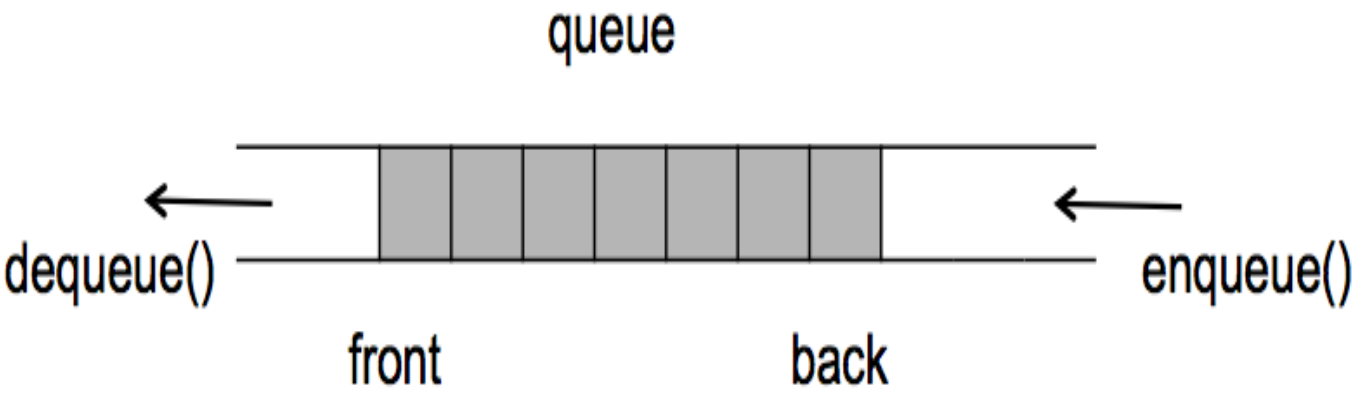
Udemy

Algorithms and Data Structures in Java

Lecture: Queue

Instructors:
George Katsilidis
Nikos Katsilidis
Christos Topalidis



What is a Queue

- A queue supports the insert and remove operations using a first-in first-out (FIFO) discipline.

Methods :

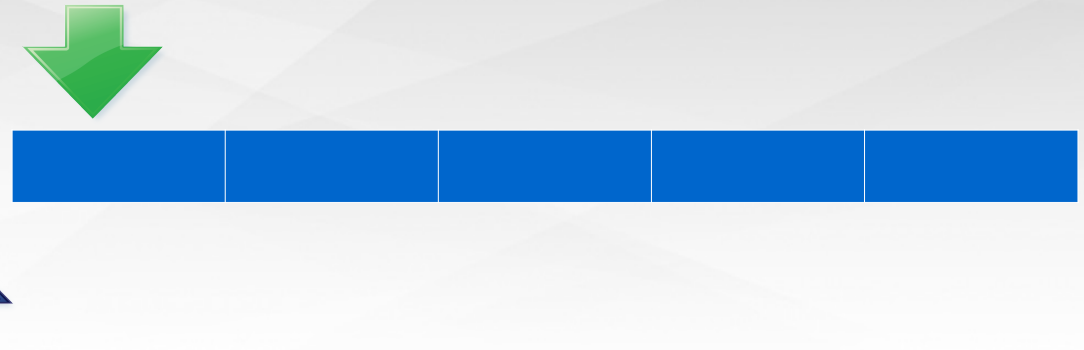
- SetSize(size)
- insert(data)
- IsEmpty()
- IsFull()
- Size()
- peek()
- removeData()

How to declare a Queue

```
public class Queues {
    int size ;
    int []Array ;
    int front = 0;
    int rear = -1;
    int itemCount = 0;
}
```

front

rear



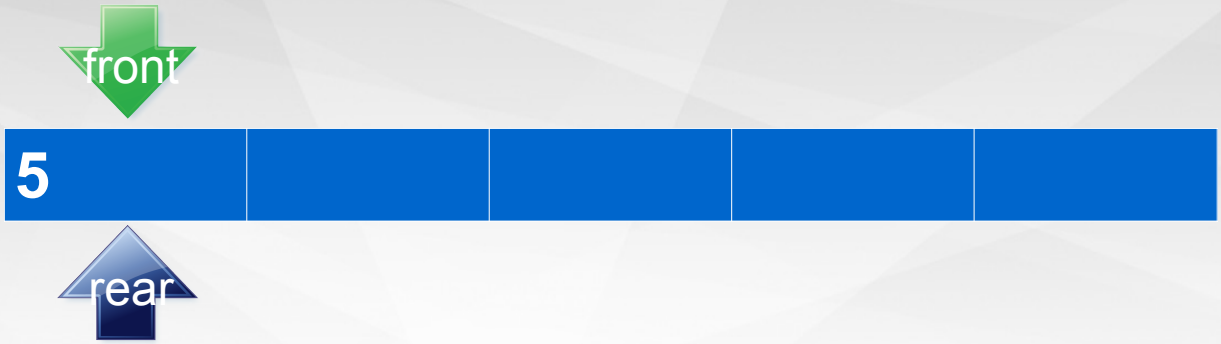
setSize

```
public class Queues {
    int size ;
    int []Array ;
    int front = 0;
    int rear = -1;
    int itemCount = 0;
    public void setSize(int size){
        this.size = size;
        Array = new int[size];
    }
}
```

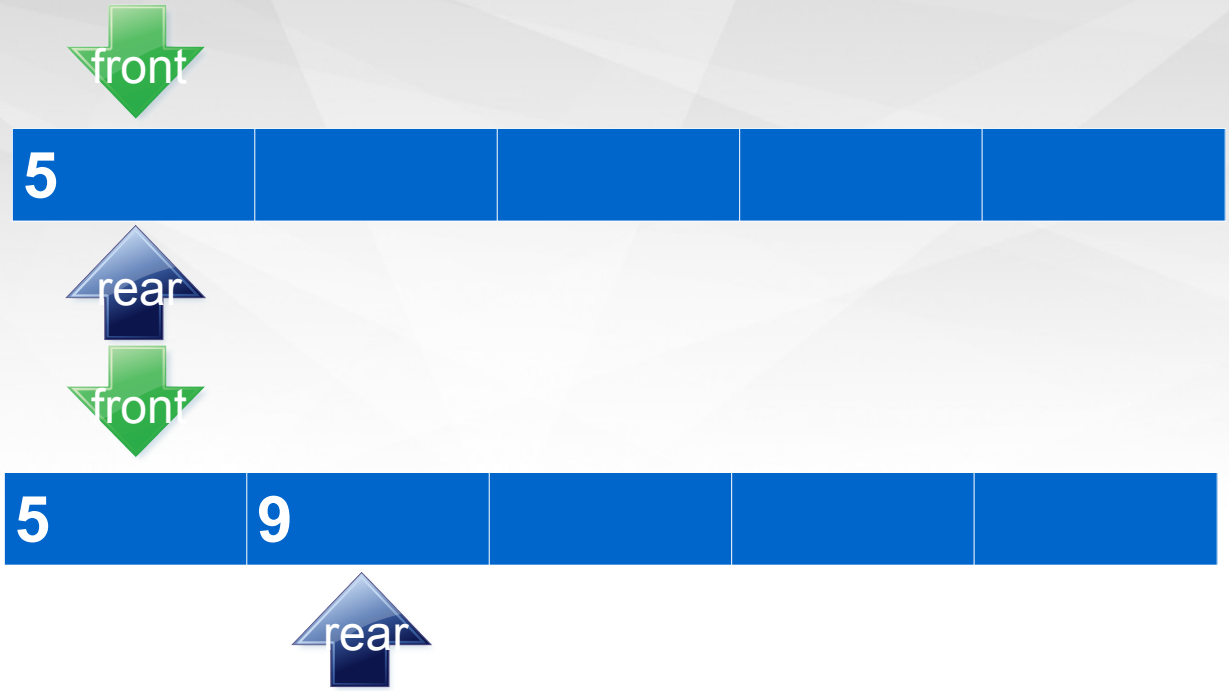
front

rear

Insert(data)



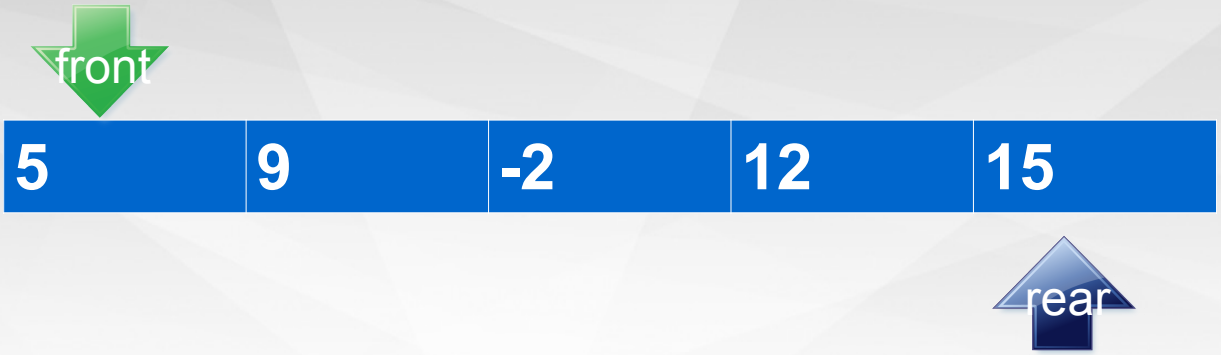
Insert(data)



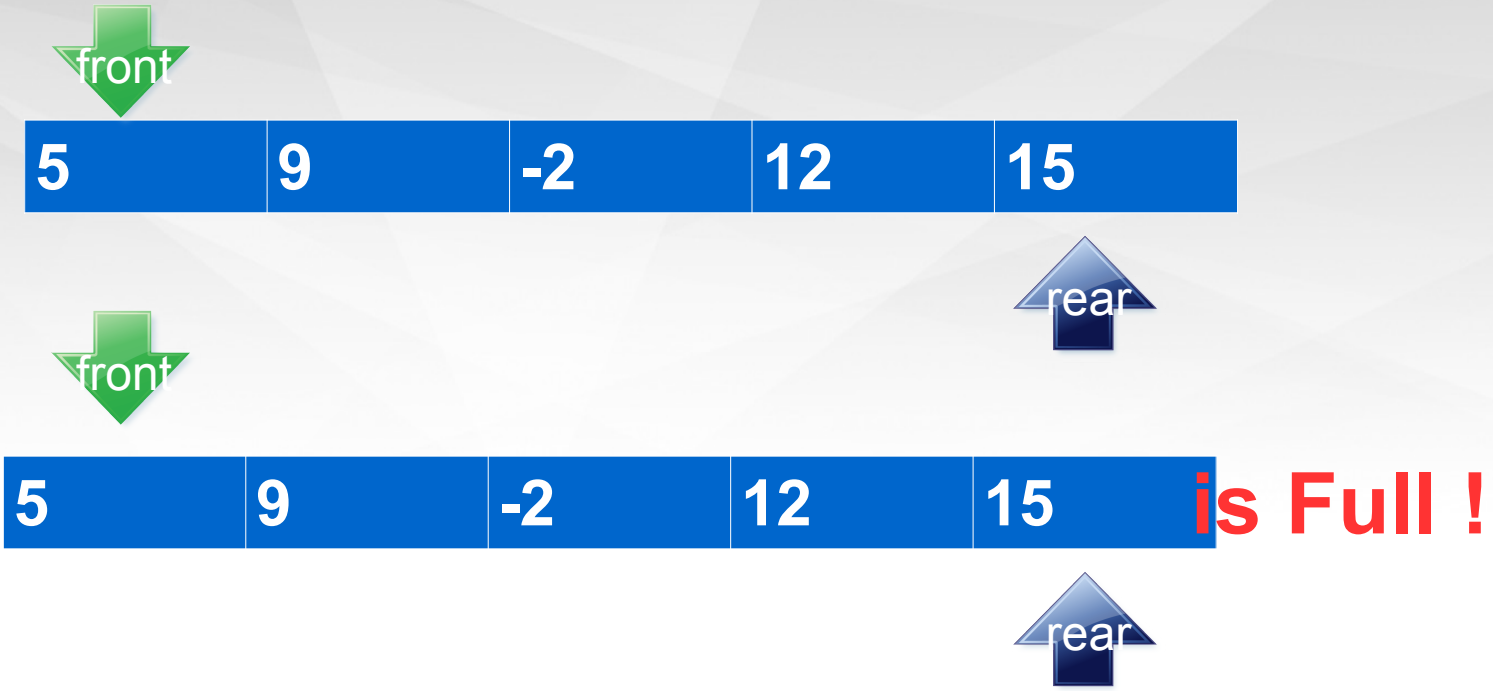
Insert(data)

Insert(data)

Insert(data)



Insert(data)



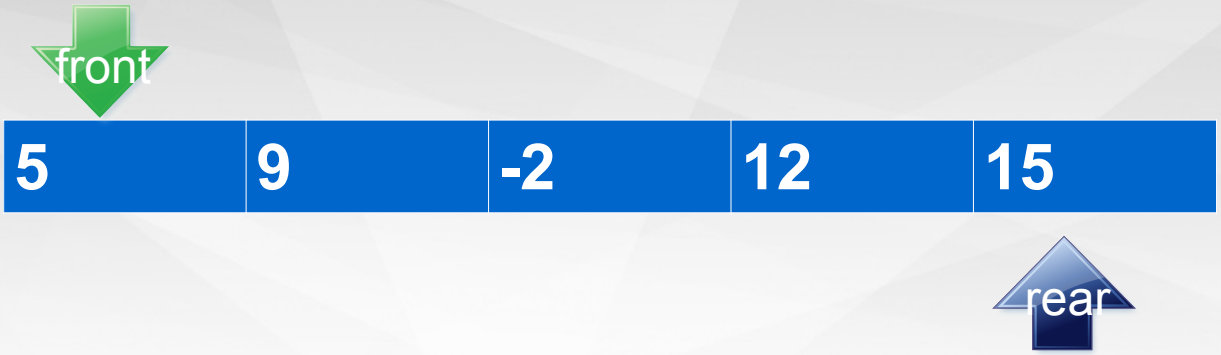
is Full !

Insert(data) Code

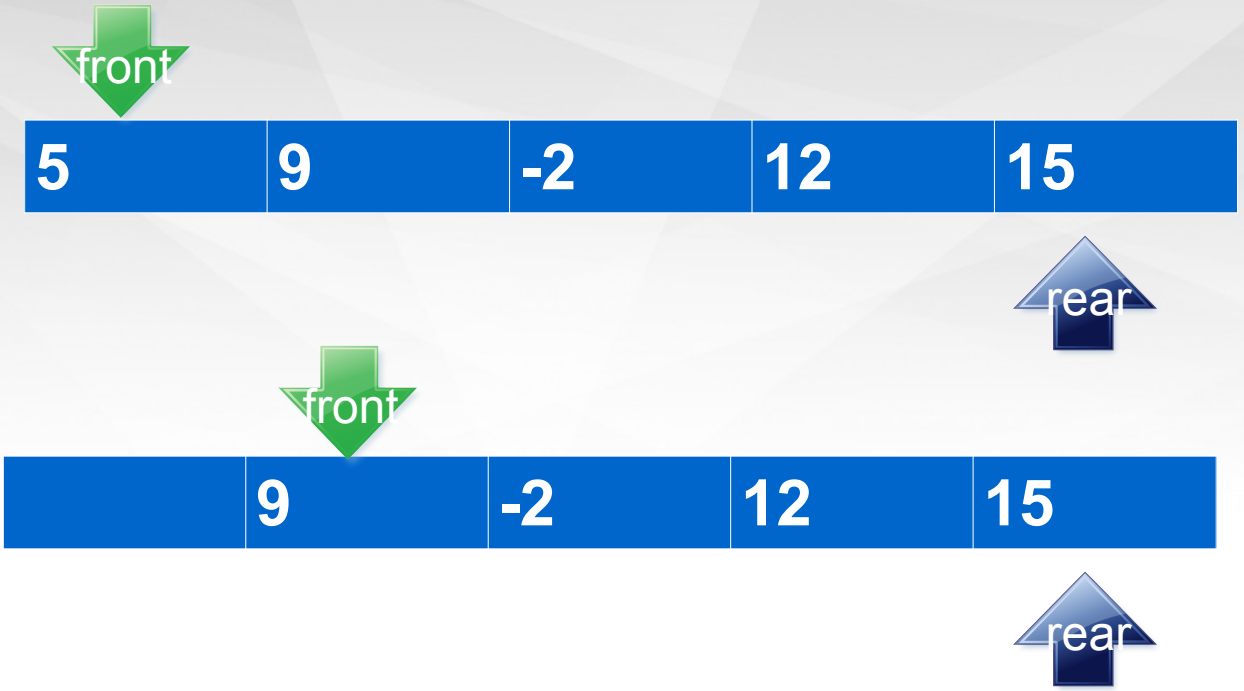
```
public void insert(int data) {
  if(!isFull()) {
     if(rear == size-1) {
        rear = -1;
      }

      Array[++rear] = data;
      itemCount++;
    }
  }
```

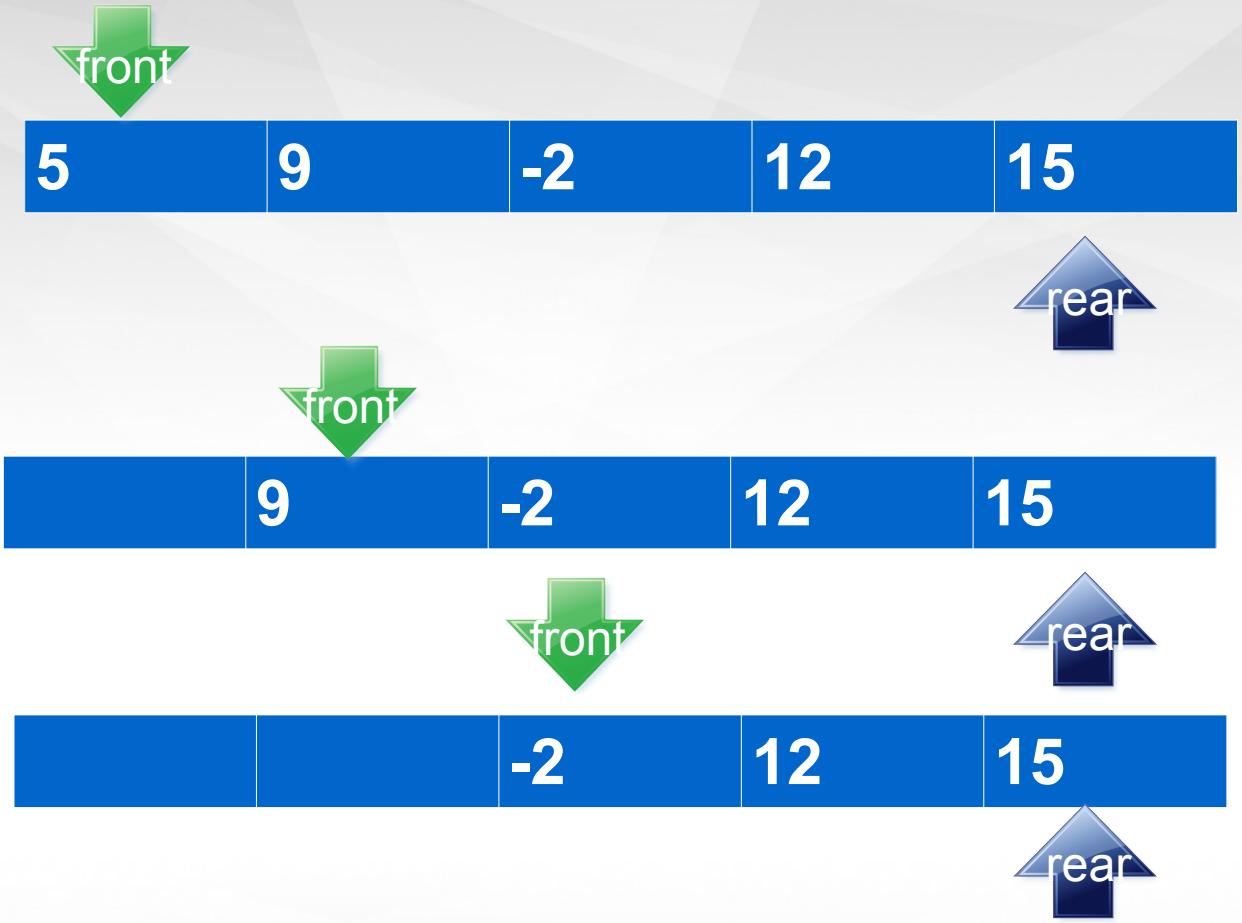
removeData()



removeData()



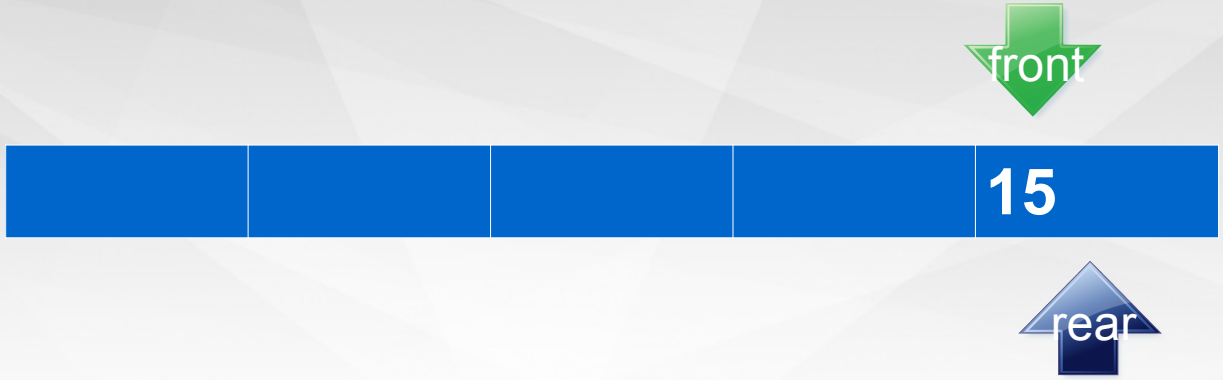
removeData()



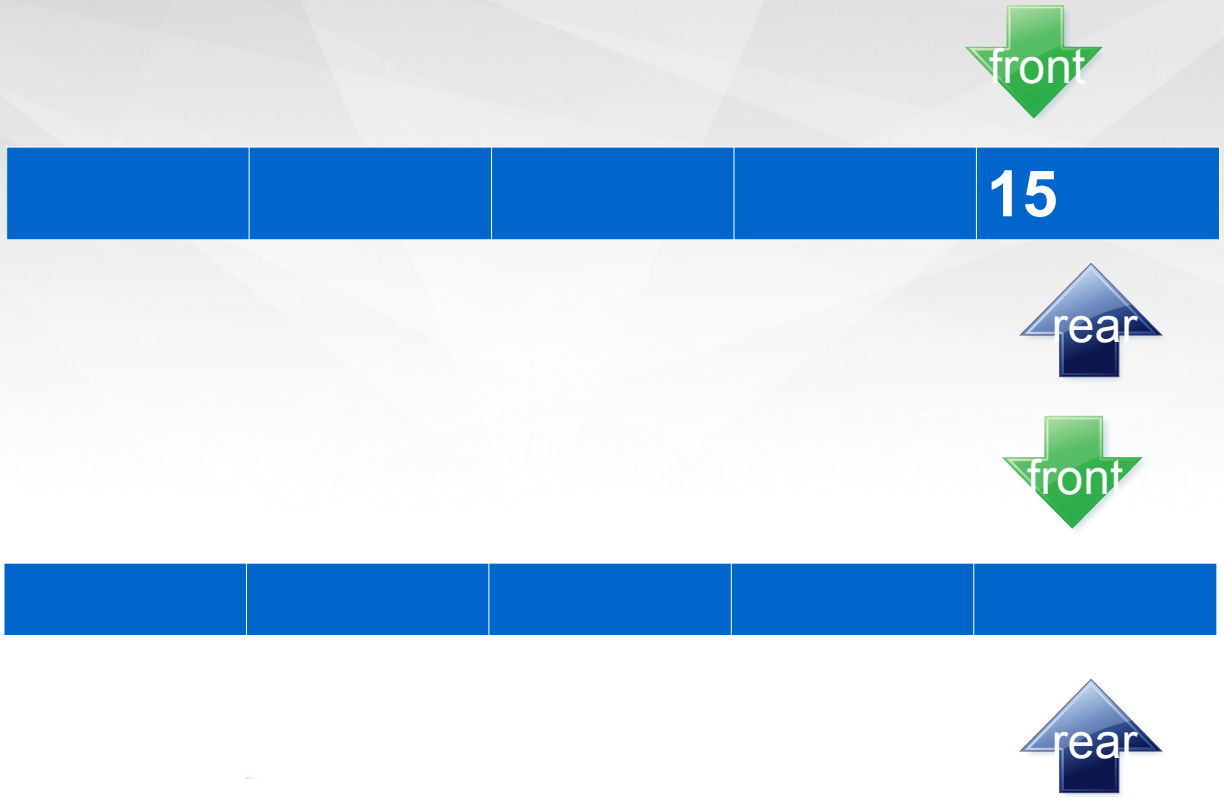
removeData()


front
5
9
-2
12
15
rear
front
9
-2
12
15
rear
front
-2
12
15
rear
front
12
15
rear

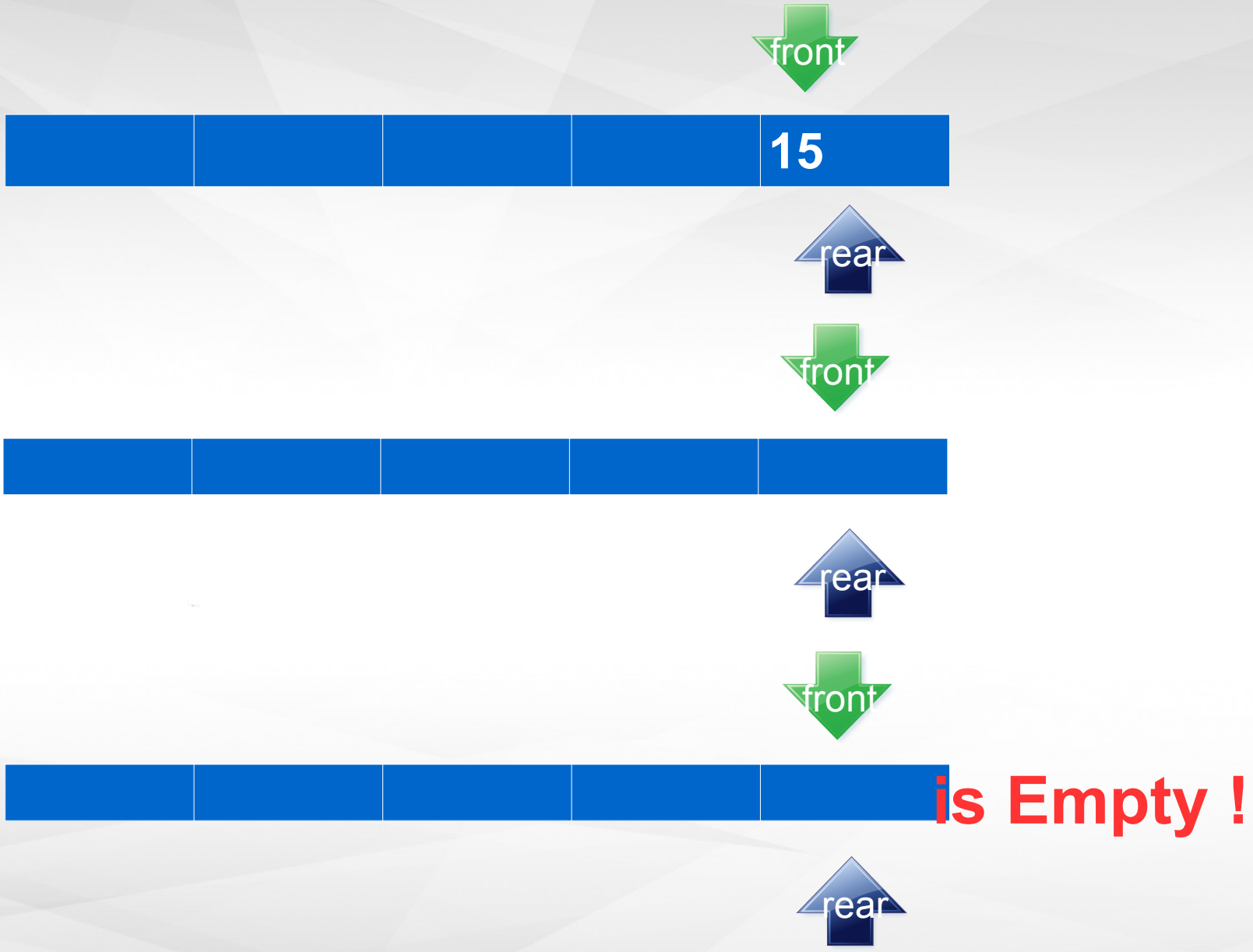
removeData()



removeData()

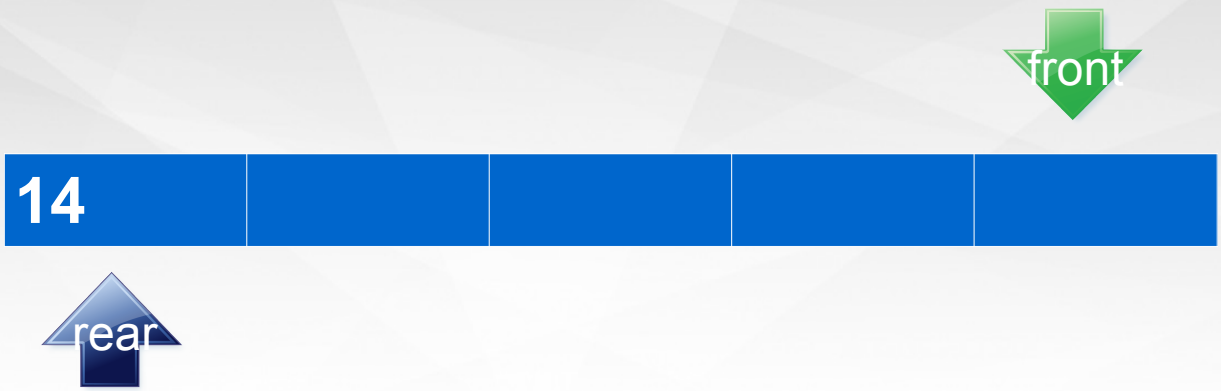


removeData()

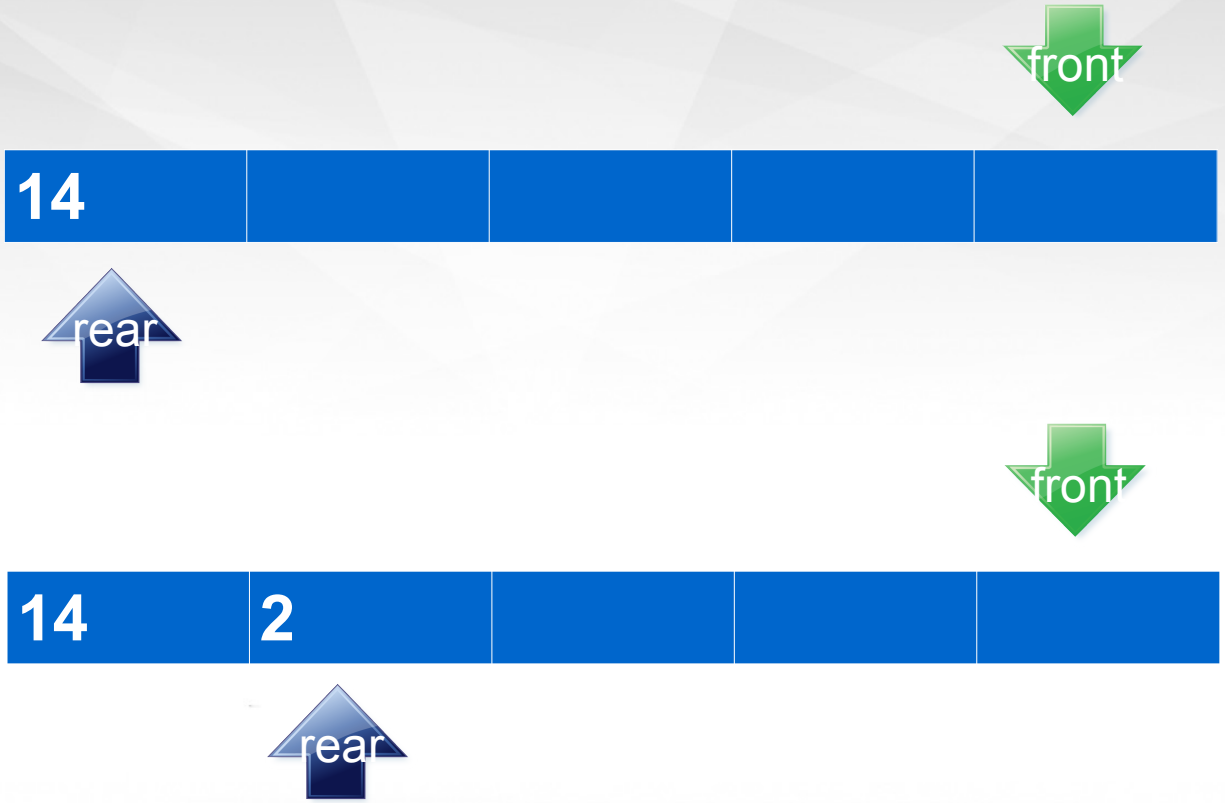


is Empty !

insert(data)



insert(data)



insert(data)

RemoveData() Code

```
public int removeData() {
      int data = Array[front++];


      if(front == size) {
         front = 0;
      }
    if(itemCount>0)
            itemCount--;
      return data;
   }
```

Methods

```
public int peek() {
    return Array[front];
}
public boolean isEmpty() {
    return itemCount == 0;
}
public boolean isFull() {
    return itemCount == size;
}
public int size() {
    return itemCount;
}
```