Queue

1). a

2). front point at 1, back point to 4 , which store new items , p

3). c

4). b

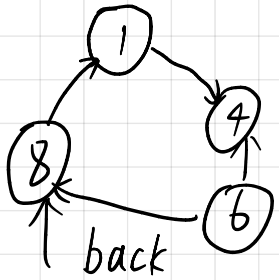
5). check if the queue is empty
return the element that front point to

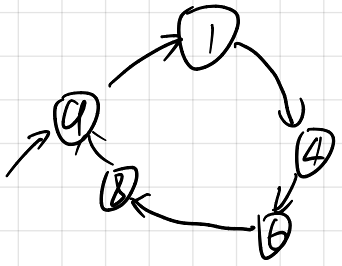
O(1)

6). add → O(n)
remove → O(1)

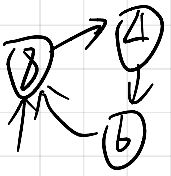
7).
```
① add        O(1)
If( check if there are empty ){
  creat a new Item   P is pointer
   p.next = back.next
   back.next = P
   back = P
} else [

}
```





```
② remove      O(1)
  If( check if there are less than 2 element) {
     val = back.next.value
     back.next = back.next.next.
     return val.
  } else {
     return a element or reture empty queue
  }
```



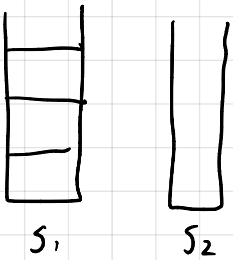
8). add() is O(n), It need look through all the list If the order of element is least.

remove is O(1), It just remove the first element regardless of the order of element.

9). change the back element as highest rank.

change the back element as a specified rank.

10).



```
stack 1
stack 2
① public void add(X) {
      while (front is not empty) {
            pop the element from front to back
      }
      front.push(X)
      while (back is not empty) {
            pop the element from back to front
      }
  }
② remove
     pop the front stack
③ peek
     return the first element of front
④ empty
     check if the front is empty.
```