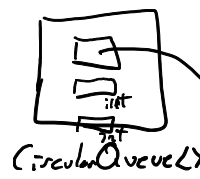
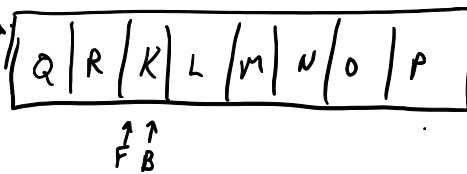
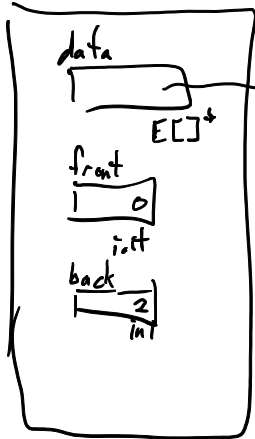
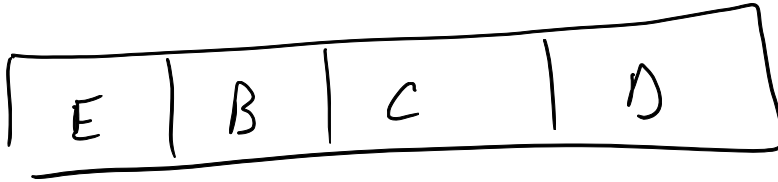


```
PureQueue<String> q = new Queue<>(8);
    ↑
Circular
```



offer
poll
peek
empty



```
public class CircularQueue<E> implements PureQueue<E> {
```

```
    private E[] data;
    private int front;
    private int back;
```

```
    public CircularQueue(int capacity) {
        data = (E[]) new Object[capacity];
        front = 0;
        back = 0;
    }
```

```
    public boolean empty() {
        return data[front] == null;
    }
```

```
    public boolean offer(E element) {
```

```
        if (element == null) {
            throw new RuntimeException("WHATEVER THIS SHOULD BE");
        }
        boolean changed = data[back] == null;
```

```
        if (changed) {
            data[back] = element;
```

```
            back++;
            if (back > data.length) {
                back = 0;
            }
        }
```

```
        back = incrementIndex(back);
```

```
    }
    return changed;
}
```

```
    O(1) public E peek() {
        return data[front];
    }
```

```
    public E poll() {
        E removed = data[front];
        if (removed != null) {
            front = incrementIndex(front);
        }
        return removed;
    }
```

O(1)

O(1)

```
private int incrementIndex(int index) {
    return (index + 1) % data.length;
}
```

O(1)

O(1)