



## 2-11 作业解析：没有 size 成员变量的循环队列

作业解析：浪费一个空间，但不使用 size 实现队列

如果我们不使用 size，也完全可以实现整个队列，但是，相应的，我们需要浪费一个空间。

下面的代码是我提供的参考代码，这个队列实现中不再用 size。

请大家注意其中注释的位置，给出了和视频中的实现不同的地方。

我的参考代码如下：

// 在这一版本的实现中，我们完全不使用size，只使用front和tail来完成LoopQueue的所有逻辑：)

public class LoopQueue implements Queue {

```
private E[] data;
private int front, tail;

public LoopQueue(int capacity){
    data = (E[])new Object[capacity + 1];
    front = 0;
    tail = 0;
}

public LoopQueue(){
    this(10);
}

public int getCapacity(){
    return data.length - 1;
}

@Override
public boolean isEmpty(){
    return front == tail;
}

@Override
public int getSize(){
    // 注意此时getSize的逻辑：
    // 如果tail >= front，非常简单，队列中的元素个数就是tail - front
    // 如果tail < front，说明我们的循环队列“循环”起来了，此时，队列中的元素个数为：
    // tail - front + data.length
    // 画画图，看能不能理解为什么？
    //
    // 也可以理解成，此时，data中没有元素的数目为front - tail，
    // 整体元素个数就是 data.length - (front - tail) = data.length + tail - front
    return tail >= front ? tail - front : tail - front + data.length;
}

@Override
public void enqueue(E e){

    if((tail + 1) % data.length == front)
        resize(getCapacity() * 2);

    data[tail] = e;
    tail = (tail + 1) % data.length;
}

@Override
public E dequeue(){
```