

用 add first 或 add last 这套新的 API 改写 Deque 的代码（第四课课件p12）：

```
我认为把push都改为addLast()/offerLast(),
pop改为removeFirst()/pollFirst()即可。
Deque<String>
deque = new LinkedList<String>();
deque.offerLast("a");
deque.offerLast("b");
deque.offerLast("c");
System.out.println(deque);
String
str = deque.peek();
System.out.println(str);
System.out.println(deque);
while
(deque.size() > 0) {
System.out.println(deque.pollFirst());
}
System.out.println(deque);
```

分析Queue:

队列就是一个先进先出的数据结构。有4个重要的method。当然略去了throw exception的方法，同样的效果只取了返回false或null的。方法的意思其实看名字就知道了。

- E remove()
- boolean offer()
- E poll()
- E peekl()

分析PriorityQueue：

优先队列和普通队列唯一的区别就是优先队列可以定义优先级。从而pop时候优先pop掉优先级最高的。默认的优先级排列是数字小的优先，ASCII码小的优先。当然可以自己定义优先级排列的方法。放入的元素必须是具有可比较属性的。重要的方法和queue一模一样。多记忆一个void clear()。

当自己定义优先级方法试，使用的语法举例 Queue<User> q = new
PriorityQueue<>(new UserComparator)); 详细代码：
<https://www.liaoxuefeng.com/wiki/1252599548343744/1265120632401152>

- 25. K个一组翻转链表
- 思路：处理LinkedList的常规思路，创建一个dummy 节点。每k个节点，断掉链表，然后只需把连接关系完全颠倒过来再连上即可。
- 流程：
 1. 创建dummy
 2. 创建四个辅助节点 pre, curr, tail,next
 3. 用while loop控制结束条件。
 4. 用for loop把tail移到正确位置。注意break的条件，别漏了一层循环。
 5. 断开链表，把tail后的断开，这样下面的函数可以终止
 6. 写个函数，把k个链表的连接关系逆转
 7. 跟新连接关系，更新pre, tail

142 环形链表II

- 思路：这题有点坑，要用数学公式的。基本思路就是龟兔赛跑，快慢指针，快指针每次跑2格，慢指针一格。通过数学公式可以证明，当快慢指针相遇后。再创造一个指针，这个指针和慢指针相遇点，即是环的入口

20. 有效的括号

思路：括号的配比就是个先进后出的问题，用栈stack非常合适。思路就是创建个栈，遇到左括号，就把后括号push进栈，看到右括号就看下栈最上面有没有和他匹配的。

155.最小栈

思路： 用一个辅助栈来记录最小值。为空时最小栈压入任何值，当有跟小的输入时，最小栈压入这个最小值。经过测试，使用deque 并不会比使用stack慢，所以，要用stack的时候用ArrayDeque替代。 Deque <> stack = new ArrayDeque<>();

deque重要的API(ArrayDeque, LinkedList相同)：
isEmpty(), offerFirst offerLast pollFirst, pollLast, peekFirst, peekLast,

84.柱状图中最大的矩形

知识点：单调栈+哨兵

思路： 用单调栈记录最大高度的index。这个index一下的都是高度比他低的，当遇到一个高度比这个高度低的，就可以求一次这个高度的面积。同时头和尾加入两个哨兵。头的哨兵确保了栈空时计算的正确。尾哨兵保证了所有的高度都会被计算到。

代码框架：

1. 特殊case处理
2. 建立新的array，头尾加入哨兵。创建单调栈，压入哨兵、
3. 遍历所有的高度，用while loop来处理遇到高度低的情况。并计算面积。

239.滑动窗口最大值

知识点：双端队列

思路： 创建双端队列后。先把已经不在窗口范围内的值移出去。然后再比较新加入的值是否比已经在stack里的大，是的话把所有小的值都移出去。最后把这个值压入栈

641.设计循环双端队列

知识点：双端队列

思路： 用一个list模拟双端队列。除了list,还要创造front, rear, size, capacity四个变量。主意front, end 减1是要考虑碰到负数。因此要(front - 1+ capacity)%capacity.

42.接雨水：

知识点：单调栈

思路：和84题十分相似。这次是单调递减。当下一根棒子比上一根高时候，就有可能接到雨水，因此对面积进行一波计算。

细节： 判断下一根棒子是不是比stack最后一根长。长的话stack出栈，并把相同长度的都出栈。

66.加一

知识点：数学

思路：纯数学逻辑。从最后一位开始判断，假如有进位则最后一位是0，判断前一位要不要进位。否则返回加一后的数组。加入所有位数都进位，则返回比原数组长度+1，第0位是1的新数组。

21合并两个有序链表

知识点：LinkedList dummy node

思路:创建一个dummy node。然后就比大小调整next。最后的小诀窍就是因为是有序数组，剩下的数组可以直接连上，不用一个一个连接。