


resize的复杂度分析

resize $O(n)$

假设当前capacity = 8，并且每一次添加操作都使用addLast

1 1 1 1 1 1 1 1 8 + 1



9次addLast操作，触发resize，总共进行了17次基本操作

resize的复杂度分析

resize $O(n)$

平均，每次addLast操作，进行2次基本操作

这样均摊计算，时间复杂度是 $O(1)$ 的！

在这个例子里，这样均摊计算，比计算最坏情况有意义。

均摊复杂度 amortized time complexity

resize $O(n)$

addLast 的均摊复杂度为 $O(1)$

同理，我们看removeLast操作，均摊复杂度也为 $O(1)$

resize的复杂度分析

resize $O(n)$

9次addLast操作，触发resize，总共进行了17次基本操作

平均，每次addLast操作，进行2次基本操作

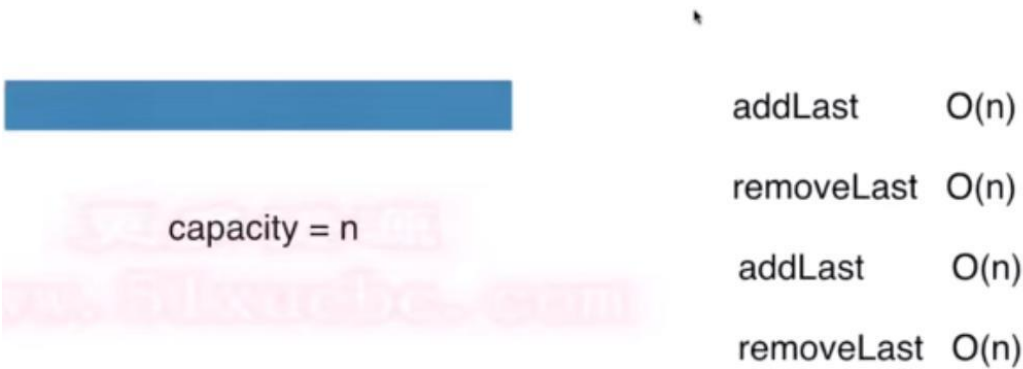
假设capacity = n，n+1次addLast，触发resize，总共进行2n+1次基

平均，每次addLast操作，进行2次基本操作

当出现复杂度左右摇摆时，应该将激进的容量调整策略改成 lazy 的策略：



但是，当我们同时看addLast和removeLast操作：



出现问题的原因：removeLast 时 resize 过于着急（Eager）

解决方案：Lazy



当 $size == capacity / 4$ 时，才将capacity减半