

# 对数器

建议：不要跳过，非常重要的自我验证技巧

## 对数器的试用场景

你在网上找到了某个公司的面试题，你想了好久，感觉自己会做，但是你找不到在线测试，你好心烦..

你和朋友交流面试题，你想了好久，感觉自己会做，但是你找不到在线测试，你好心烦..

你在网上做笔试，但是前几个测试用例都过了，突然一个巨大无比数据量来了，结果你的代码报错了，如此大的数据量根本看不出哪错了，甚至有的根本不提示哪个例子错了，怎么debug？你好心烦…

# 对数器

## 对数器的实现

- 1, 你想要测的方法a（最优解）
- 2, 实现复杂度不好但是容易实现的方法b（暴力解）
- 3, 实现一个随机样本产生器（长度也随机、值也随机）
- 4, 把方法a和方法b跑相同的输入样本，看看得到的结果是否一样
- 5, 如果有一个随机样本使得比对结果不一致，打印这个出错的样本进行人工干预，改对方法a和方法b
- 6, 当样本数量很多时比对测试依然正确，可以确定方法a（最优解）已经正确。

关键是第5步，找到一个数据量小的错误样本，便于你去带入`debug`

然后把错误例子带入代码一步一步排查

`Print`大法、断点技术都可以

对数器的门槛其实是比较高的，因为往往需要在两种不同思路下实现功能相同的两个方法，暴力一个、想象中的最优解是另一个。

以后的很多题目都会用到对数器，几乎可以验证任何方法，尤其在验证贪心、观察规律方面很有用

到时候会丰富很多对数器的实战用法，这里只是一个简单易懂的示例