6.2.3 当路途不再平坦

肯尼迪总统的父亲约瑟夫.P.肯尼迪据说是这么教育他的孩子的：“当路途不再平坦，天堑就开始化为通途”。但是他从来都没有给软件纠错过。这个段子确实对于在给软件纠错过程中遇到困难的时候如何去做包含有一些相当实用的提示。

1.找那些平常的怀疑点。比如，你是否：

(1)把针对某些功能的内容提要的顺序搞错了

(2)犯了名词的拼写错误。比如当需要输入大写字母时误写为小写。

(3)没有重新启动变量

(4)测试时把两个浮点数弄成相等（==）而不是几乎相等（要记住浮点数算法和你在学校学的算术是不一样的。

(5)当试图测试对象相等（比如id（L1）==id(L2)）时弄成数值相等(比如写出L1==L2来比较两个列表)

(6)忘记了某些内置功能存在副作用。

(7)忘记了（）会把对一个函数类型的引用变成调用。

(8)创建了一个未定义的对象。或

(9)犯了其他你常犯的错误。

2.别再问自己这程序为什么不按照你想的去运行。而是去问自己它为什么会像现在这样来运行。那也许更容易回答，而且可能会成为在找出修复程序办法过程中的一个好的开端。

3.牢记于心：bug很可能并不在你以为的地方。如果在，那你很可能早就修复了。一种寻找bug的实用方法是找它最不可能在哪里。正如夏洛克福尔摩斯所说：“排除掉其他因素，留下的一定就是真相。”

4.试试看把问题讲给别人听。每个人都存在盲点。很多时候仅仅是试图把问题讲给别人就可以让你看见你忽视的东西。一个很好的解释的点就是bug为什么不在某个地方。

5.不要看见什么信什么，尤其是不要相信文档记载。你的代码也许不会照着评论意见那样运行。

6.不要再查bug了，开始写文档记载。这也许能从不同的角度发现问题。

7.走开，然后明天再试试。这也许表示这bug解决的要比你钻牛角尖完成要来的晚一些，但是也许你不用花那么多时间来找它。那也许是拿延迟换效率（学生们，这就是为什么对于编程问题晚做不如早做）。

6.2.4. 当你找到了那个Bug

当你觉得你已经找到了你代码中的bug，几乎难以抵抗立刻开始编码和测试修复的诱惑。然而，最好稍微慢下脚步等一等。记住目标不是修复一个错误，而是迅速而高效的以没有bug的程序前进。

问问你自己这个bug是不是解释了出现的所有问题，还是说只是冰山一角。如果是后者，那最好是在处理bug的同时考虑其他的变化。比如，假如你发现bug是一个偶然变化的列表的后果，你可以局部绕过这个问题（也许用拷贝列表的方法），或者你可以考虑用元组来代替列表（因为元组是不可变的），也许可以消除掉这个程序中别的地方的bug。

在做出修改之前，尽量了解这种改变可能带来的衍生物。它会不会破坏其他地方？会不会导致程序过于复杂？它是否提供了一种机会来整理代码的其余部分？

永远确保自己能够回到现在的这个状况。没有什么，比在做出一系列的改变之后却发现自己比一开始离目标更远，并且无法复原，更让人心生绝望的。硬盘空间总是够用的，用它来放置你的程序的旧版本。

最后，如果存在很多难以解释的错误，你也许应该考虑一次找到并修复一个bug是不是还是正确的。也许你最好考虑一下有没有一种更好的组织程序的方法，或者一些能使程序更好更简单实现的算法。

附：本翻译完全原创。文中“bug”一词因其在该语境下”小错误”的意思无歧义且被广泛接受，故文中没有翻译而使用原文。

03014436 王卓立