Black-box testing

黑箱测试

- Test suite designed without looking at code
  在不看代码的情况下设计的测试套装
  - Can be done by someone other than implementer
    制定人，完成者
  - Will avoid inherent biases of implementer, exposing potential bugs more easily
    能够避免完成者的内在偏见
    更容易地暴露潜在bugs
  - Testing designed without knowledge of implementation, thus can be reused even if implementation changed
    在不知道实施细节时设计的测试
    即使实施改变的情况下，可以被拒绝

Paths through a specification

通过一个特性参数的道路

```
def sqrt(x, eps):
    """Assumes x, eps floats
            x >= 0
            eps > 0
       returns res such that
            x-eps <= res*res <= x+eps"""
```

- Paths through specification:
  - x = 0
  - x > 0
- But clearly not enough

Paths through a specification

- Also good to consider boundary cases
  经常最好去考虑边界条件
  - For lists: empty list, singleton list, many element list
    空列表 单元素列表 多元素列表
  - For numbers, very small, very large, “typical”
    对于数字 非常小 非常大 典型

Example

- For our sqrt case, try these:
  - First four are typical
    - Perfect square
    - Irrational square root
    - Example less than 1
  - Last five test extremes
    - If bug, might be code, or might be spec (e.g. don't try to find root if eps tiny)

x	eps
0.0	0.0001
25.0	0.0001
.05	0.0001
2.0	0.0001
2.0	1.0/2.0**64.0
1.0/2.0**64.0	1.0/2.0**64.0
2.0**64.0	1.0/2.0**64.0
1.0/2.0**64.0	2.0**64.0
2.0**64.0	2.0**64.0