

软件测试

(Software Testing)

outline

- 软件测试基础
- 软件测试策略
- 软件测试技术

软件测试概述

- 软件工程的其它阶段

- 构造出系统

- 软件测试

- 证明程序中有错误

“建设性”

“破坏性”

软件测试概述

- Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use.
- When you test software, you execute a program using **artificial data**.
- Can reveal the **presence** of errors NOT their **absence**.
- Testing is part of a more general **verification and validation** process, which also includes static validation techniques.

软件测试的目标

- 测试是为了发现程序中的错误而执行程序的过程；
- 好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案；
- 成功的测试是发现了至今为止尚未发现的错误的测试。

----G. Myers

软件测试准则

- 所有测试都应该能追溯到用户需求
 - 最严重的错误是导致程序不能满足用户需求的那些错误
- 应该远在测试开始之前就制定出测试计划
 - 完成了需求模型就可以制定测试计划
- 把Pareto原理应用到软件测试中
 - 测试发现的错误中80%很可能是由程序中20%的模块造成的

软件测试准则

- 应该从“小规模”测试开始，并逐步进行“大规模”测试
 - 单个程序模块 → 集成的模块簇 → 整个系统
- 穷举测试是不可能的
 - 测试只能证明程序中有错误，不能证明程序中没有错误
- 为了达到最佳的测试效果，应该由独立的第三方从事测试工作
 - 开发软件的软件工程师不是完成全部测试工作的最佳人选

测试方法

黑盒测试(功能测试):

- 把程序看作一个黑盒子;
- 完全不考虑程序的内部结构和处理过程;
- 是在程序接口进行的测试。

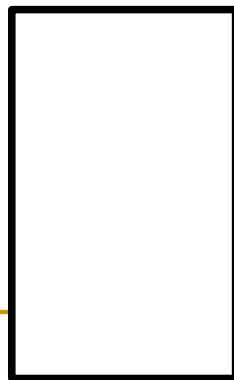
白盒测试(结构测试):

- 把程序看成装在一个透明的盒子里;
- 测试者完全知道程序的结构和处理算法;
- 按照程序内部的逻辑测试程序, 检测程序中的主要执行通路是否都能按预定要求正确工作。

测试方法

黑盒测试不可能实现穷尽测试：

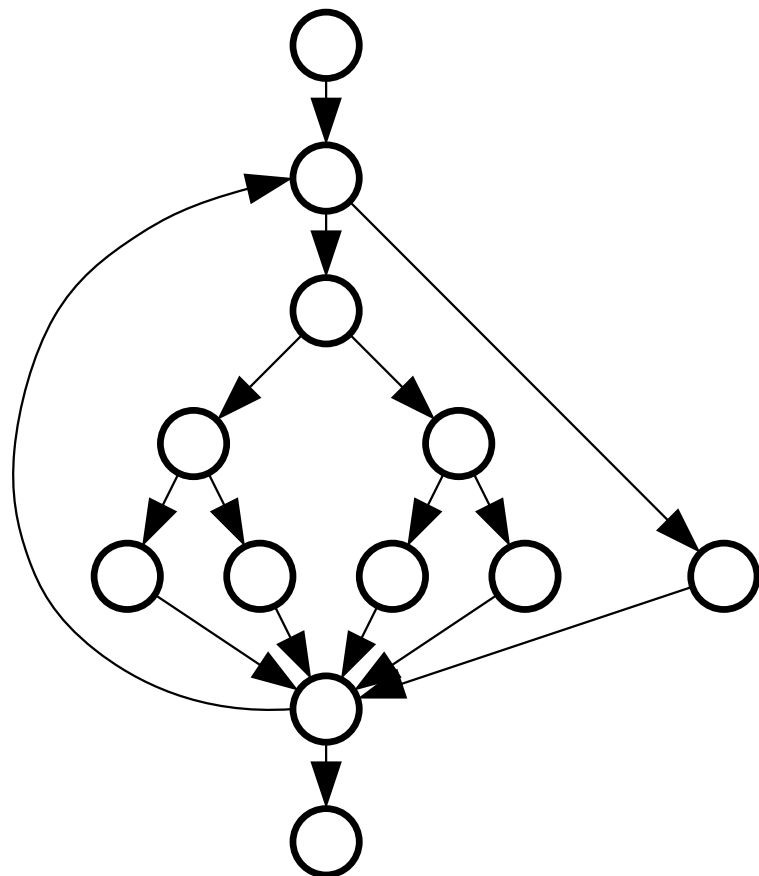
- 假设有程序P，输入量为A和B，输出量为C。
- 如果计算机的字长为32位，A和B的数据类型都是整数类型。将A和B的可能取值进行排列组合，输入数据的可能性有： $2^{32} \times 2^{32} = 2^{64}$ 种。
- 假设这个程序执行一次需要1毫秒，要完成所有的测试，计算机需要连续工作5亿年。



测试方法

白盒测试也不能实现穷尽测试：

- 图中所示的一个小程序的控制流程。曲线代表执行次数不超过20的循环，循环体中共有5条通路。
- 可能执行的路径有 5^{20} 条，近似为 10^{14} 条可能的路径。
- 如果完成一个路径的测试需要1毫秒，那么整个测试过程需要3170年。



	黑盒测试	白盒测试
优点	①适用于各阶段测试 ②从产品功能角度测试 ③容易入手生成测试数据	①可构成测试数据使特定程序部分得到测试 ②有一定的充分性度量手段 ③可获较多工具支持
缺点	①某些代码得不到测试 ②如果规格说明有误，则无法发现 ③不易进行充分性测试	①通常不易生成测试数据 ②无法对未实现规格说明的部分进行测试 ③工作量大，通常只用于单元测试，有应用局限
性质	一种确认技术，回答“我们在构造一个正确的系统吗？”	一种验证技术，回答“我们在正确地构造一个系统吗？”