

微课



# 逻辑覆盖

主讲人：景国良

学院：计算机科学与工程学院



# 逻辑覆盖

主要内容:

- ◆ 概念
- ◆ 语句覆盖
- ◆ 判定覆盖
- ◆ 条件覆盖



把程序看成装在一个透明的白盒子里，测试者完全知道程序的结构和处理算法。

这样的测试方法我们称之为白盒测试技术。



## 微软Exchange 2000和Windows 2000中的人员结构

	Exchange 2000	Windows 2000
项目经理	25人	约 250人
开发人员	140人	约 1700人
测试人员	350人	约 3200人
开发人员/测试人员	2 : 5	约 1 : 2



# 心理对比

由于测试的目标是**暴露程序中的错误**，从心理学角度看，由程序的编写者自己进行测试是不恰当的。

希望证明程序  
“对不对”



设计者

希望证明程序  
“错没错”



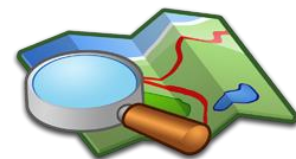
测试者



因为测试的目标是**暴露程序中的错误**，所以**测试**是为了发现程序中的错误而执行程序的过程。

白盒测试最困难的问题是**设计测试用的输入数据**。

**不可能进行穷尽的测试。**





# 穷举测试不可行

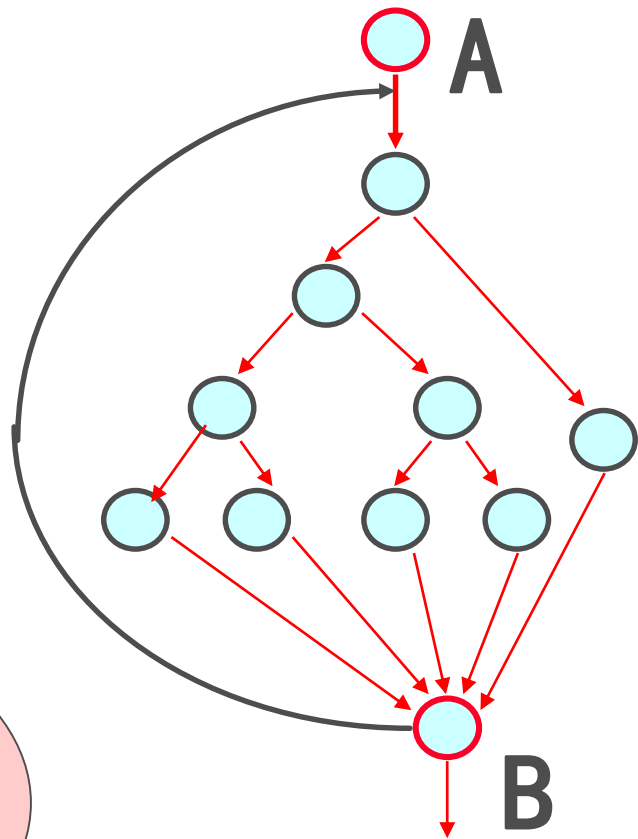
穷举测试实例：

设程序含4个分支，循环次数  
 $\leq 20$ ，从A到B的可能路径

$$= 5^1 + 5^2 + \dots + 5^{19} + 5^{20}$$
$$\approx 10^{14}$$

执行时间：

穷举测试需  
3170年以上。



# 逻辑覆盖

有选择地执行程序中某些最有代表性的通路是对穷尽测试的惟一可行的替代办法，这就是**逻辑覆盖**。

选用少量“**最有效的**”测试数据





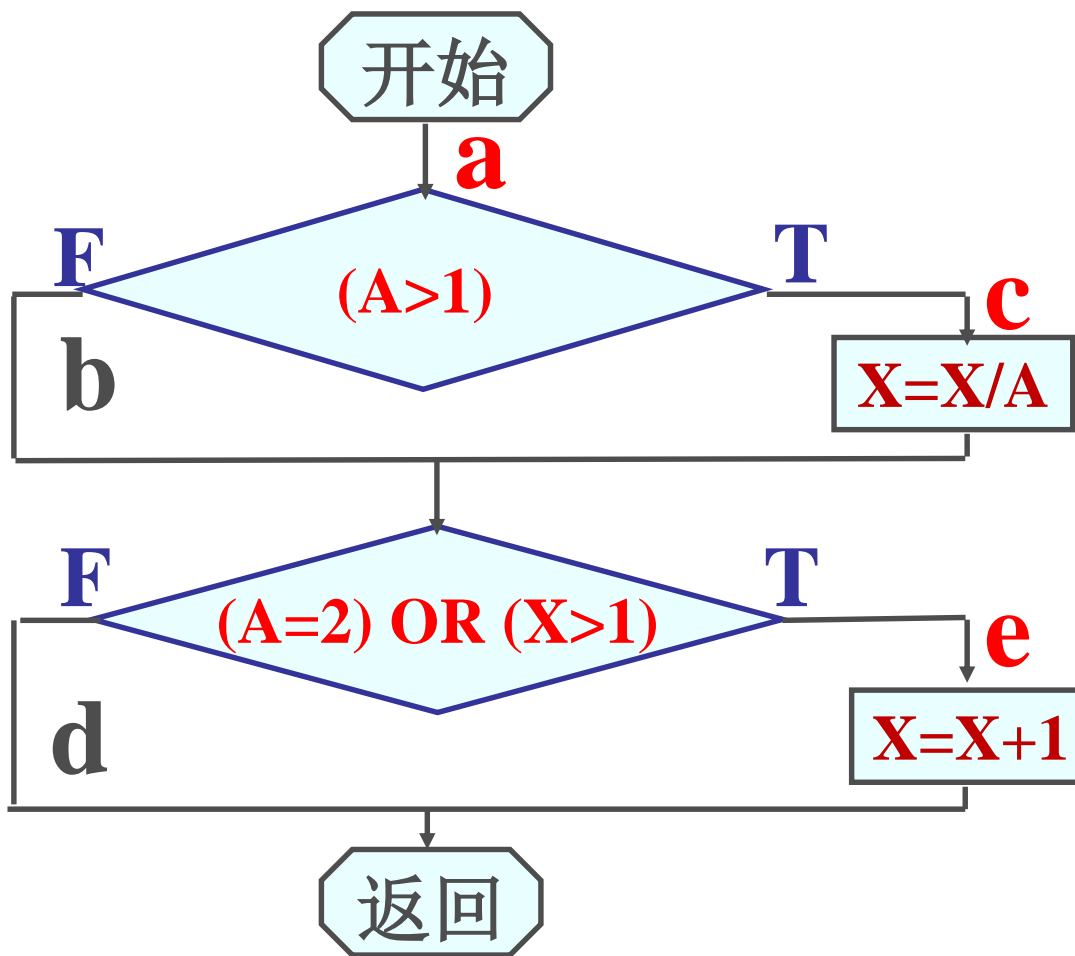
# 逻辑覆盖—语句覆盖

使程序中每个语句至少执行一次。

只需设计一个测试用例：

输入数据： $A=2$ ， $X=4$ ，

即达到了语句覆盖。



语句覆盖是最弱的

逻辑覆盖

(如： $X > 1$  写成  $X < 1$ ，查不出来)



# 逻辑覆盖—判定覆盖

使每个判定的真假分支都至少执行一次。

可设计两组测试用例：

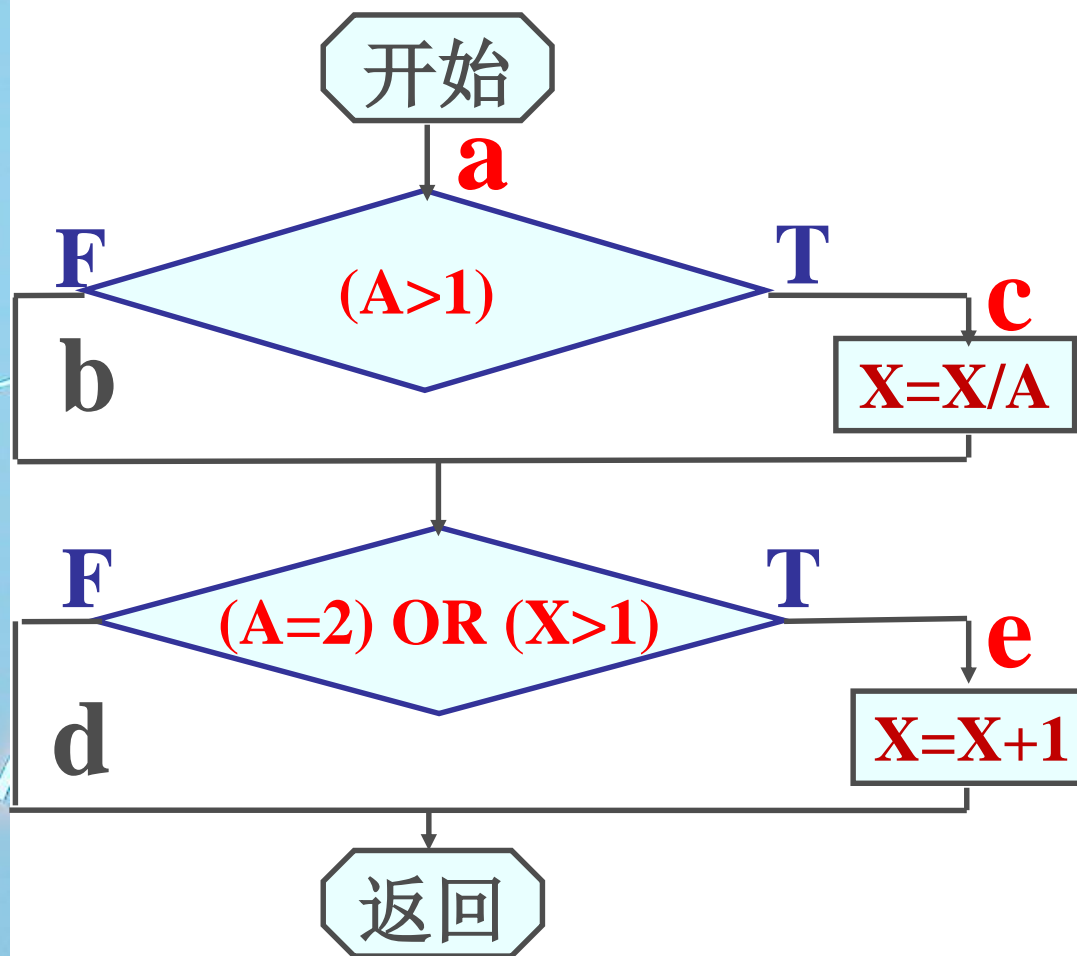
$A=3, X=3$

可覆盖  $c$ 、 $d$  分支

$A=1, X=2$

可覆盖  $b$ 、 $e$  分支

两组测试用例可覆盖所有判定的真假分支

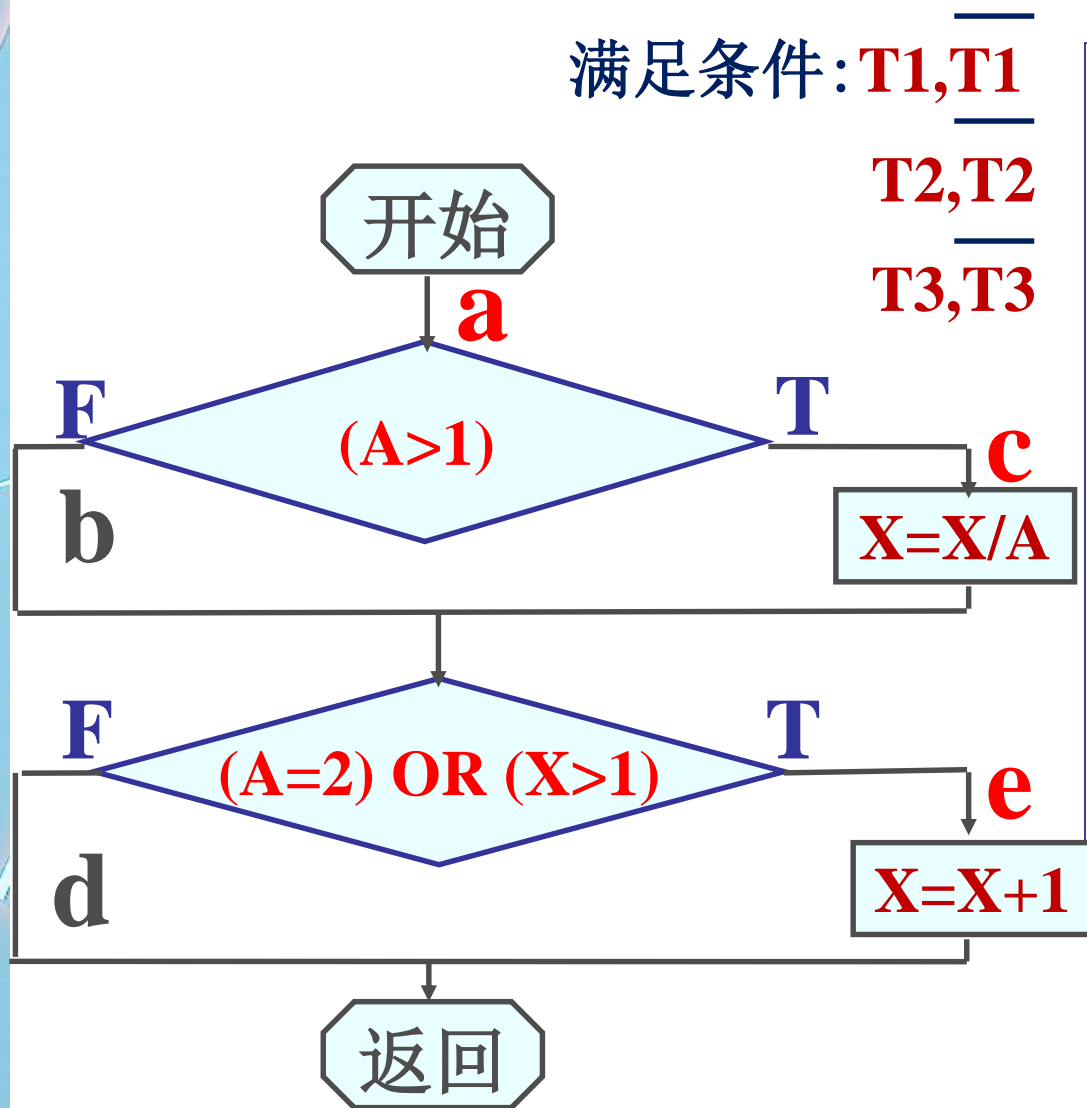


判定覆盖仍是弱的逻辑覆盖，只覆盖了全部路径的一半。



# 逻辑覆盖—条件覆盖

使每个判定的每个条件的可能取值至少执行一次。



第一判定表达式:

设条件  $A > 1$  取真 记为  $\overline{T1}$   
 假  $T1$

第二判定表达式:

设条件  $A = 2$  取真 记为  $\overline{T2}$   
 假  $T2$

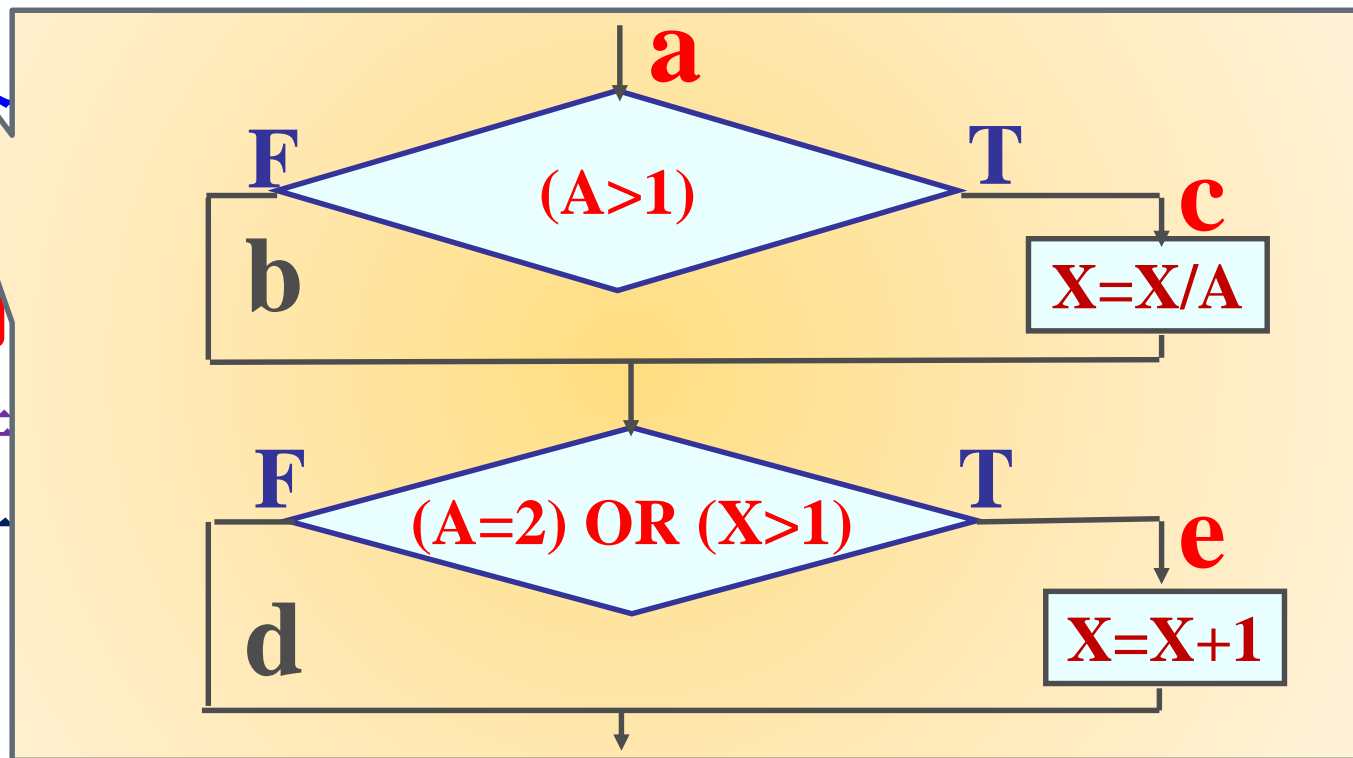
条件  $X > 1$  取真 记为  $\overline{T3}$   
 假  $T3$



# 逻辑覆盖—条件覆盖

测试用例	通过 路径	满足的 <u>条件</u>	覆盖 分支
A X 1 3	abe	T1, T2, <u>T3</u>	b, e
2 1	ace	T1, T2, <u>T3</u>	c, e

两  
课内  
条件  
不一



# 微课总结

白盒测试技术

针对语句测试的  
语句覆盖

针对判定式测试  
的判定覆盖

穷举测试的不可  
行性

针对判定条件测试的  
条件覆盖





逻辑覆盖

# 相关微课内容



相关微课：  
逻辑覆盖进一步研究



# 课后练习

- 设计下列伪码的语句覆盖、判定覆盖、条件覆盖测试用例。（参考答案见PPT最后一页）

**INPUT(A,B,C)**

**IF A>5 THEN X=10**  
**ELSE X=1**

**END IF**

**IF B>10 THEN Y=20**  
**ELSE Y=2**

**END IF**

**IF C>15 THEN Z=30**  
**ELSE Z=3**

**END IF**

**PRINT(X,Y,Z)**

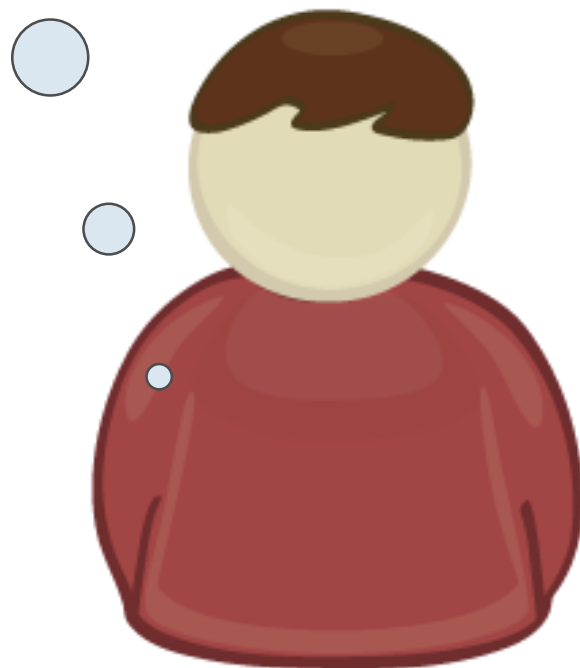




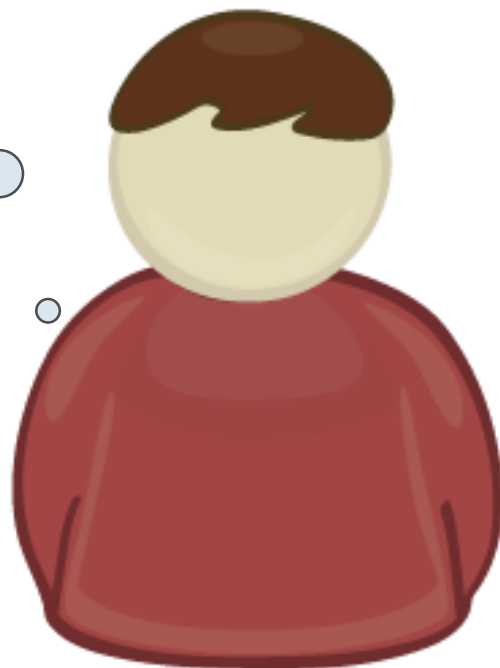
# 暂停和思考

如有问题，可将问题发  
送至邮箱：

[jingguoliang@sohu.com](mailto:jingguoliang@sohu.com)



谢谢各位老师  
与同学！



## 课后练习

### ● 参考答案

**解答：**因为每个判定里只有一个条件，因此条件的真假即为判定的真假，所以测试数据**如果实现了条件覆盖，则必然实现判定覆盖，也必然实现了语句覆盖。**实现的典型测试用例如下：

**测试数据1： A=1, B=1, C=1**

**X=1, Y=2, Z=3**

**测试数据2： A=20, B=40, C=60**

**X=10, Y=20, Z=30**

