北京理工大学

智能时代的软件测试

4.1 覆盖率测试

刘辉 教授

目录

CONTENTS

04

条件覆盖

■ 条件覆盖

- 选取足够多的测试数据，使被测试程序中每个判定表达式中的每个条件都取到各种可能的结果。

04 条件覆盖

```
float Compute(float A,float B,float X)
{
    if (A>1)&&(B==0) X=X/A;
    if (A==2)||(X>1) X=X+1;
    return X;
}
```

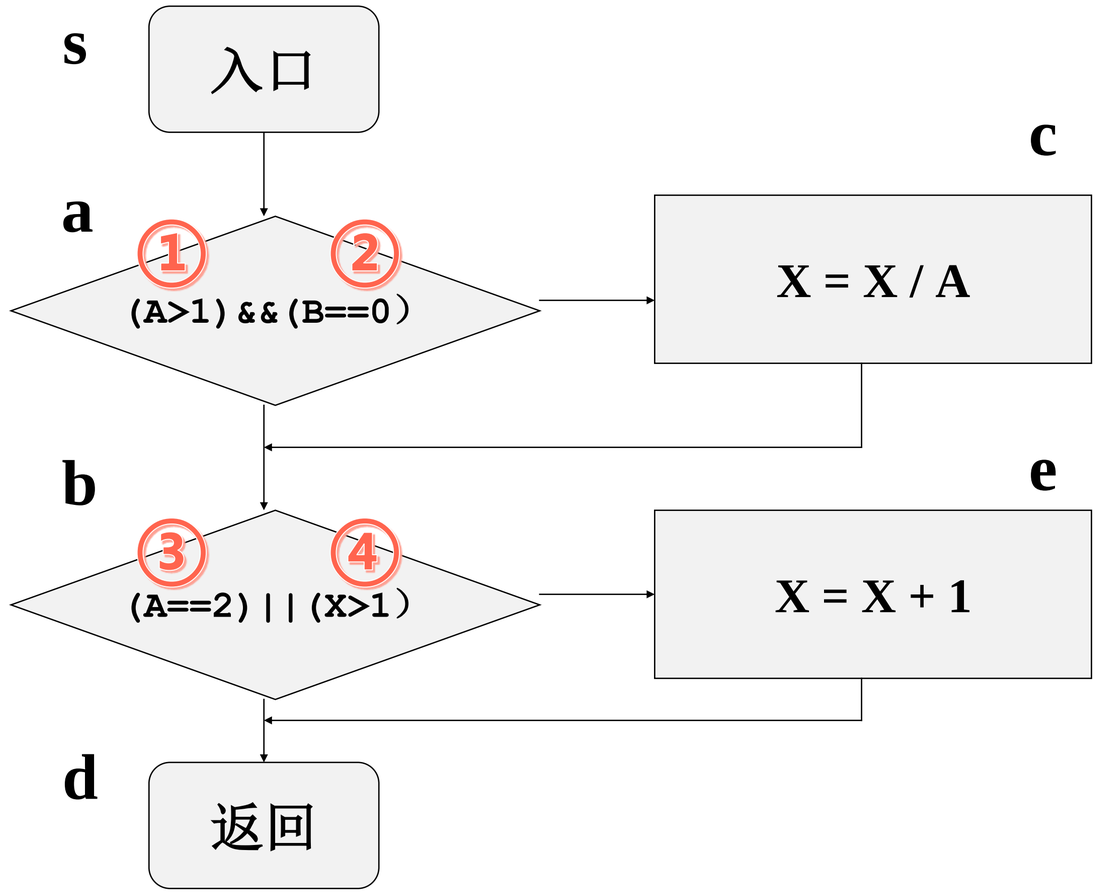
04 条件覆盖

```
float Compute(float A,float B,float X)

{
        ①      ②
    if (A>1)&&(B==0) X=X/A;

    if (A==2)||(X>1) X=X+1;
        ③      ④
    return X;

}
```



04

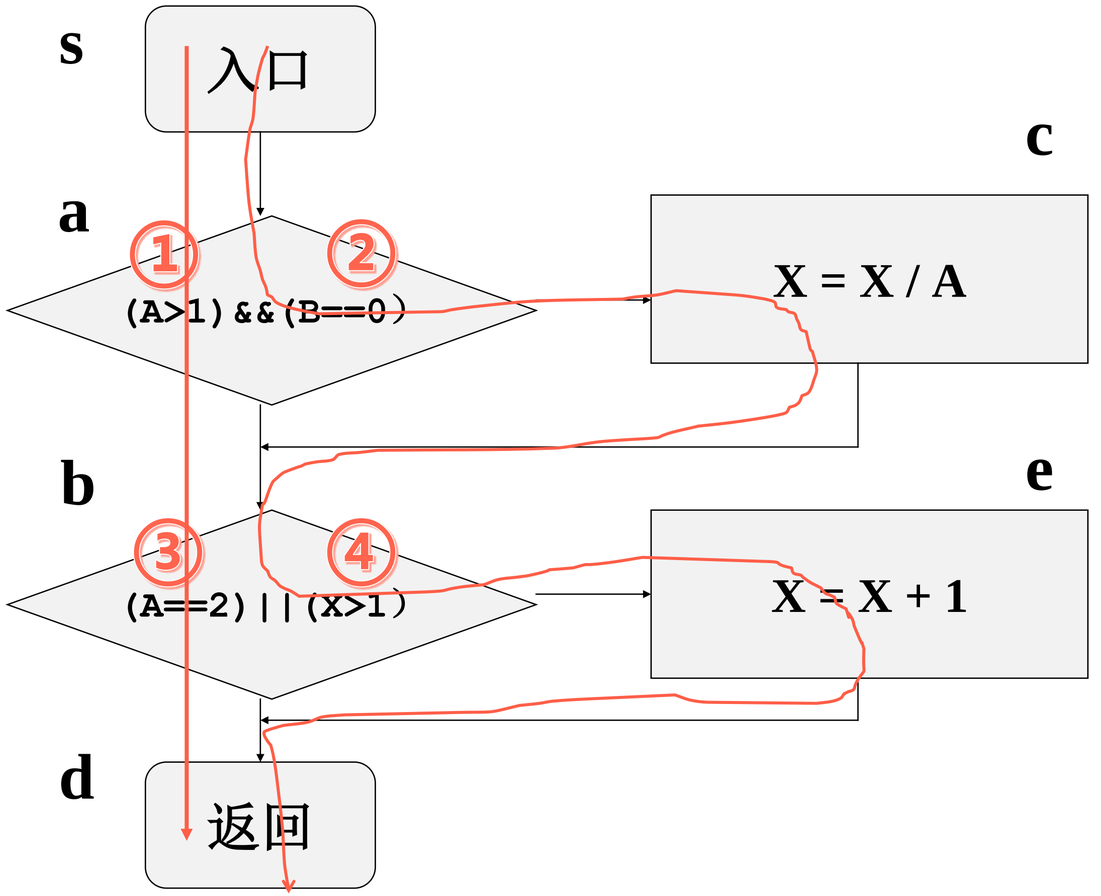
条件覆盖

■ 测试用例

ID	A	B	X
1	2	0	4
2	0	1	1

■ 测试用例执行路径

- sacbed
- sabd



04 条件覆盖

```java
package dubo;

public class Comparator {

    public int min(int x, int y, int z) {
        int min = x;
        if (y < min)
            min = y;
        if (z < min)
            min = z;
        return min;
    }
}

```

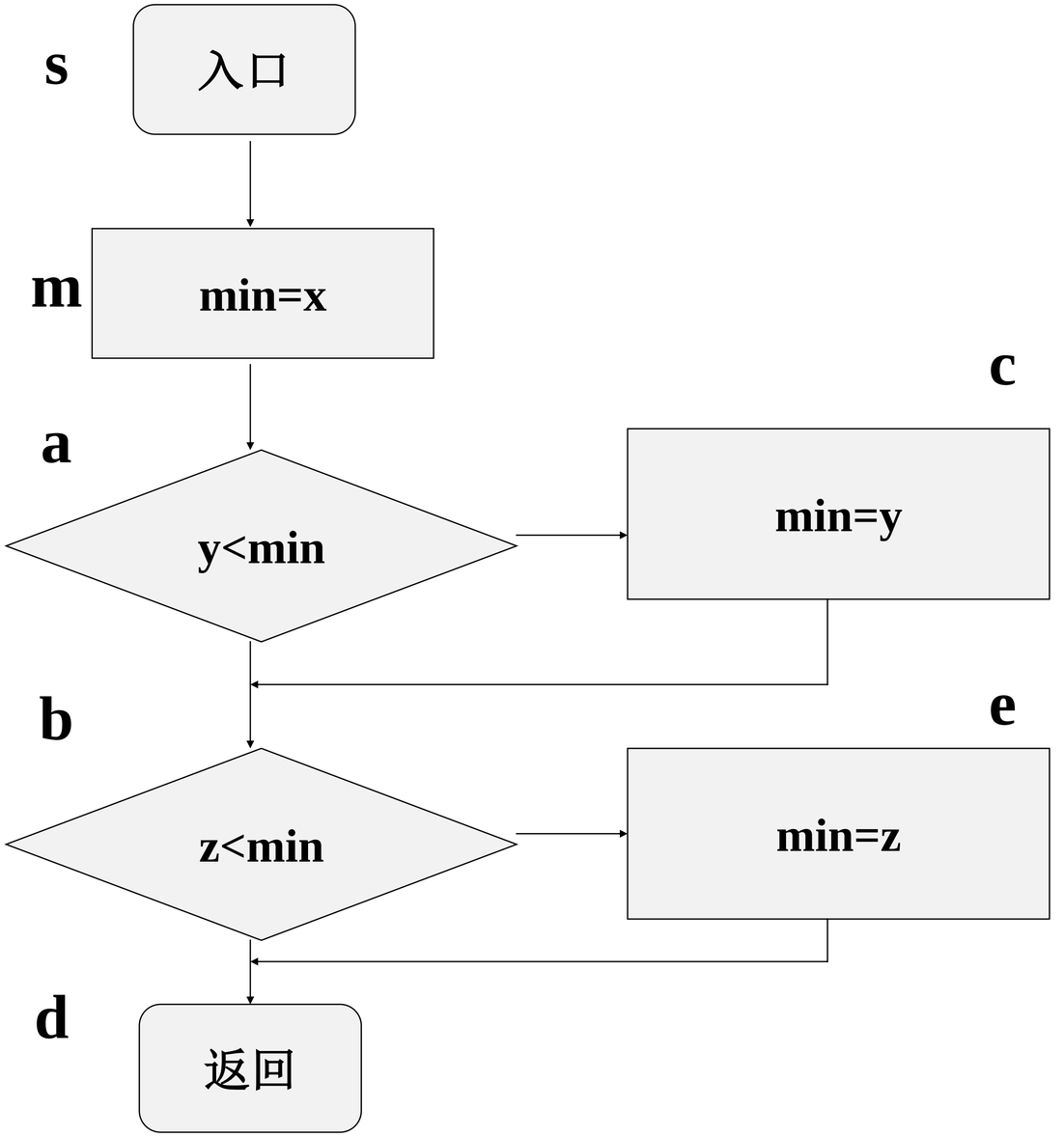
04 条件覆盖

```java
package dubo;

public class Comparator {

    public int min(int x, int y, int z) {
        int min = x;
        if (y < min)
            min = y;
        if (z < min)
            min = z;
        return min;
    }
}

```



04

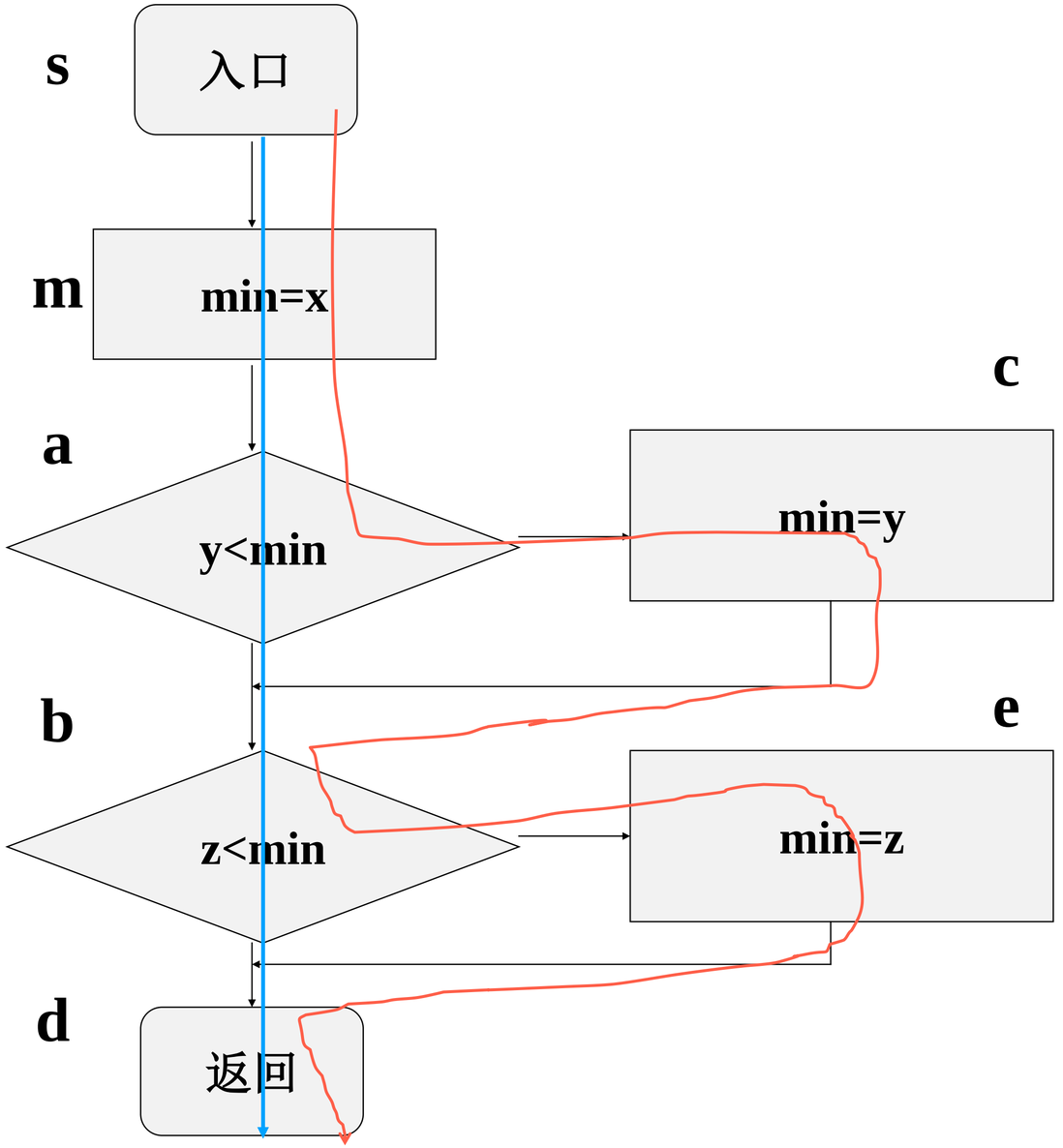
条件覆盖

■ 测试用例

ID	x	y	z
1	9	8	7
2	7	8	9

■ 测试用例执行路径

- smacbed
- smabd



04 条件覆盖

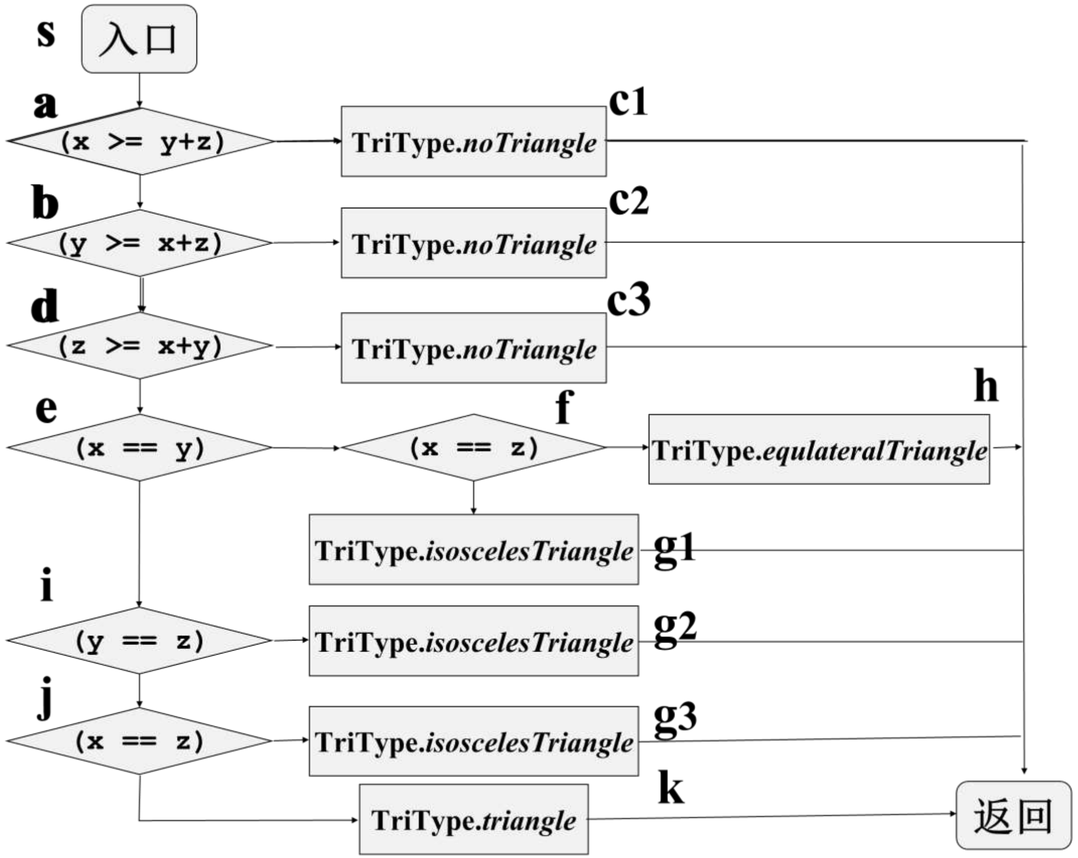
```
1  package dubo;
2  public class Triangle {
3      public TriType ReturnTriangleType(int x, int y, int z) {
4          if (x >= y + z)
5              return TriType.noTriangle;
6          if (y >= x + z)
7              return TriType.noTriangle;
8          if (z >= x + y)
9              return TriType.noTriangle;
10         if (x == y)
11             if (x == z)
12                 return TriType.equilateralTriangle;
13             else
14                 return TriType.isoscelesTriangle;
15         else if (y == z)
16             return TriType.isoscelesTriangle;
17         else if (x == z)
18             return TriType.isoscelesTriangle;
19         else
20             return TriType.triangle;
21     }
22 }
23 enum TriType {
24     equilateralTriangle, isoscelesTriangle, triangle, noTriangle
25
26 }
```

04

条件覆盖

■ 测试用例

ID	x	y	z
1	9	1	1
2	1	9	1
3	1	1	9
4	1	1	1
5	2	2	3
6	3	2	2
7	2	3	2
8	2	3	4



目录

CONTENTS

05

条件/判定覆盖

■ 条件覆盖

- 选取足够多的测试数据，使被测试程序中每个判定表达式中的每个条件都取到各种可能的结果。

05 条件/判定覆盖

■ 分支/判定覆盖

- 选取足够多的测试数据，使被测试程序中不仅每个语句至少执行一次，而且每个判定的每种可能的结果都至少执行一次。

05 条件/判定覆盖

```
float Compute(float A,float B,float X)
{
    if (A>1)&&(B==0) X=X/A;
    if (A==2)||(X>1) X=X+1;
    return X;
}
```

05

条件/判定覆盖

■ 条件/判定覆盖

- 选取足够多的测试数据，使得判定表达式中的每个条件都取到各种可能的结果，**而且**每个判定表达式也都取到各种可能的结果。
- 条件/判定覆盖==条件覆盖+判定覆盖

05 条件/判定覆盖

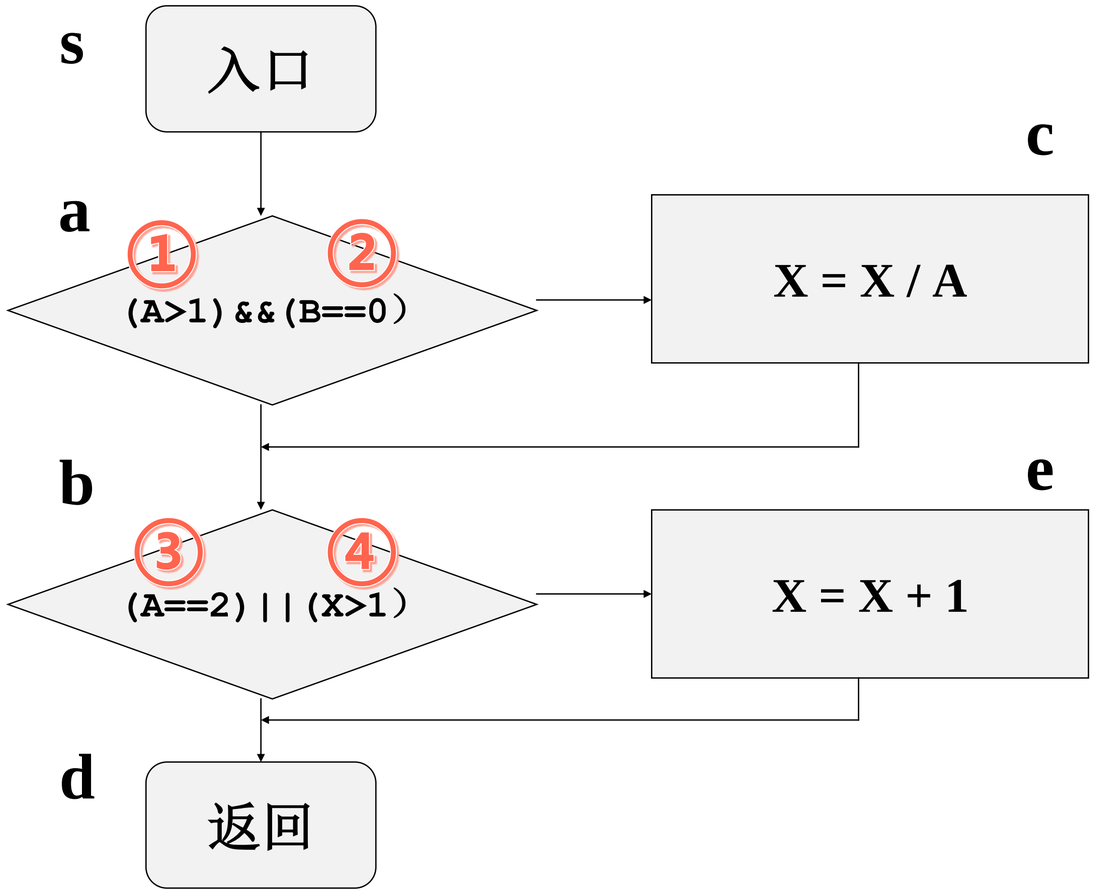
```
float Compute(float A,float B,float X)

{
      if (A>1)&&(B==0) X=X/A;

      if (A==2)||(X>1) X=X+1;

      return X;

}
```



05

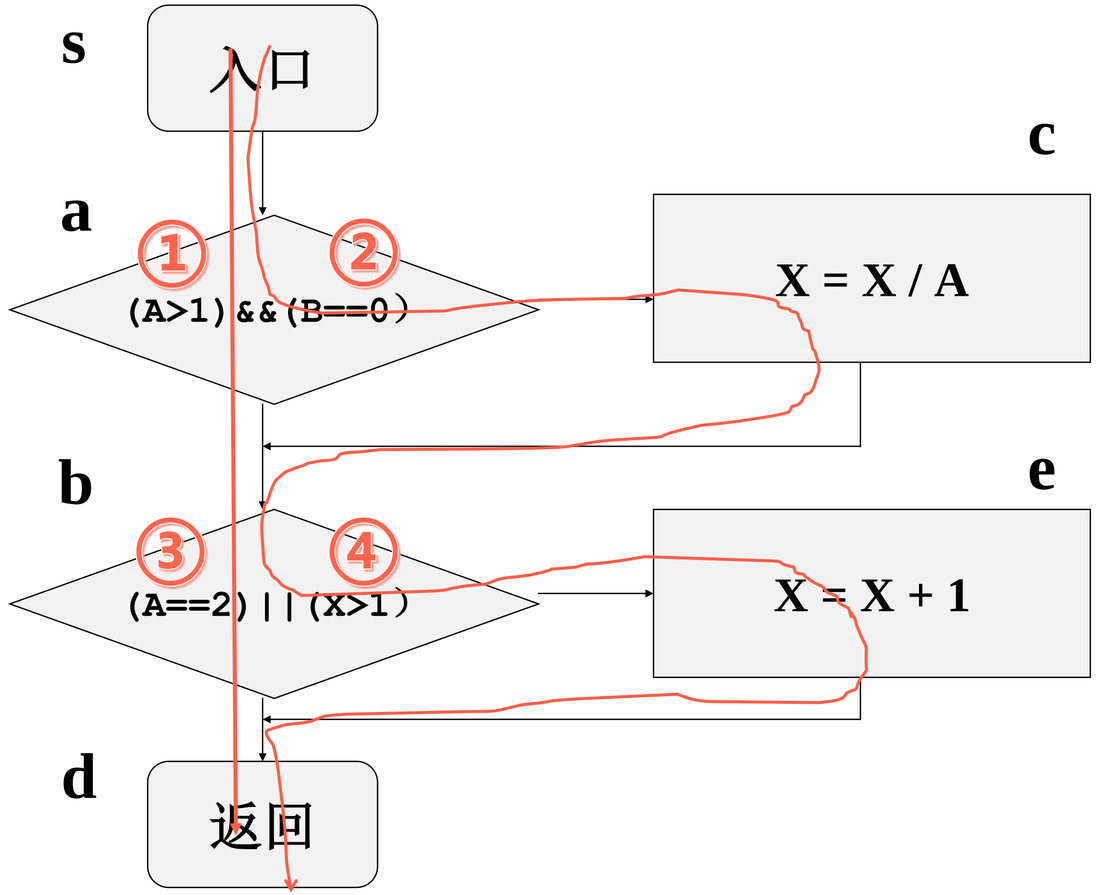
条件/判定覆盖

■ 测试用例

ID	A	B	X
1	2	0	4
2	0	1	1

■ 测试用例执行路径

- sacbed
- sabd



05

条件/判定覆盖

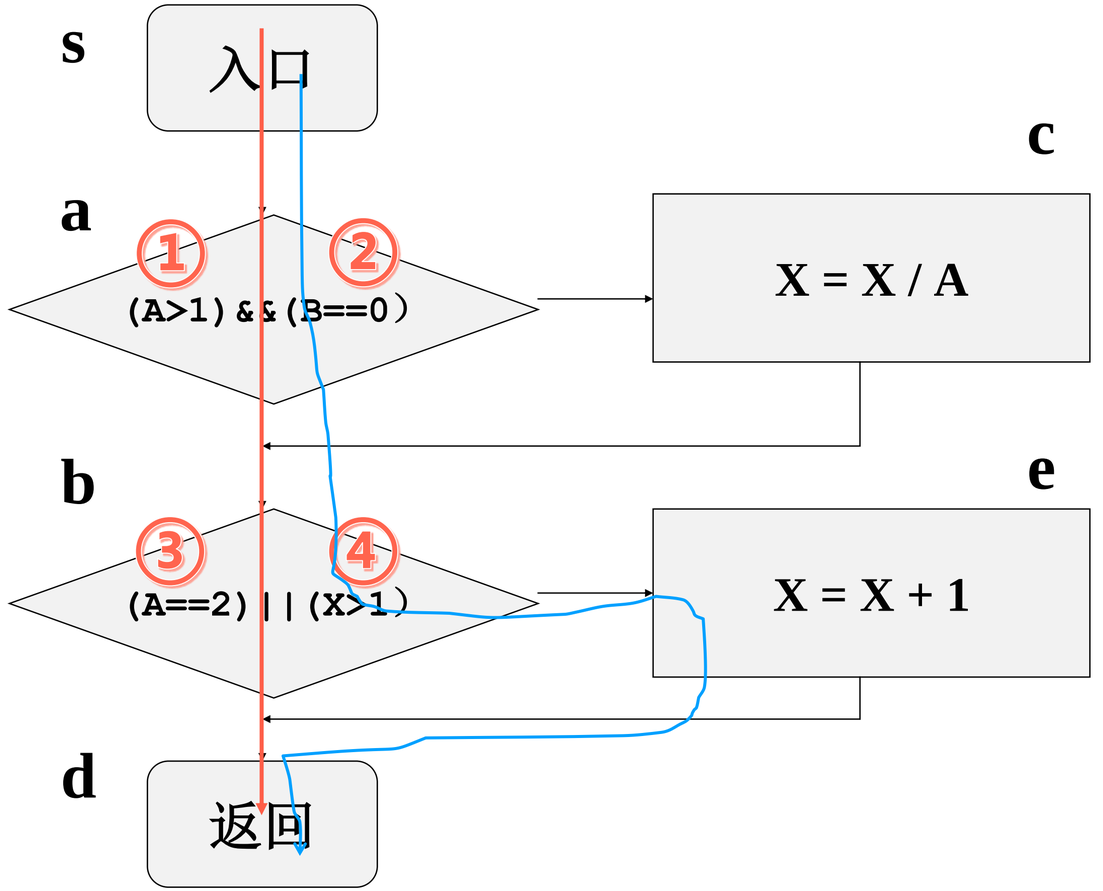
■ 测试用例

ID	A	B	X
1	2	1	4
2	0	0	1

■ 测试用例执行路径

- sabed [真假真真]
- Sabd [假真假假]

条件覆盖但是判定不覆盖的例子?



05

条件/判定覆盖

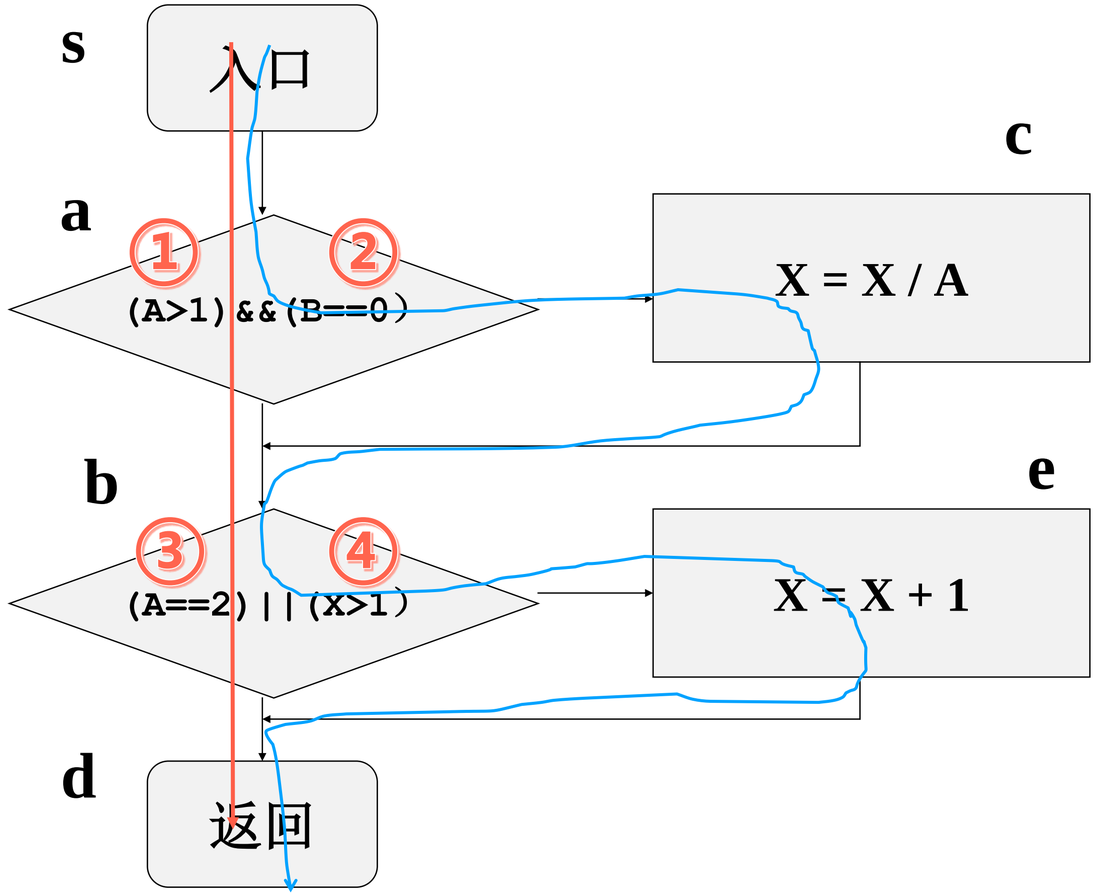
■ 测试用例

ID	A	B	X
1	2	0	4
2	3	1	2

■ 测试用例执行路径

- sabed [真真真真]
- Sabd [真假假假]

判定覆盖但是条件不覆盖的例子?



05 条件/判定覆盖

```java
package dubo;

public class Comparator {

    public int min(int x, int y, int z) {
        int min = x;
        if (y < min)
            min = y;
        if (z < min)
            min = z;
        return min;
    }
}

```

05

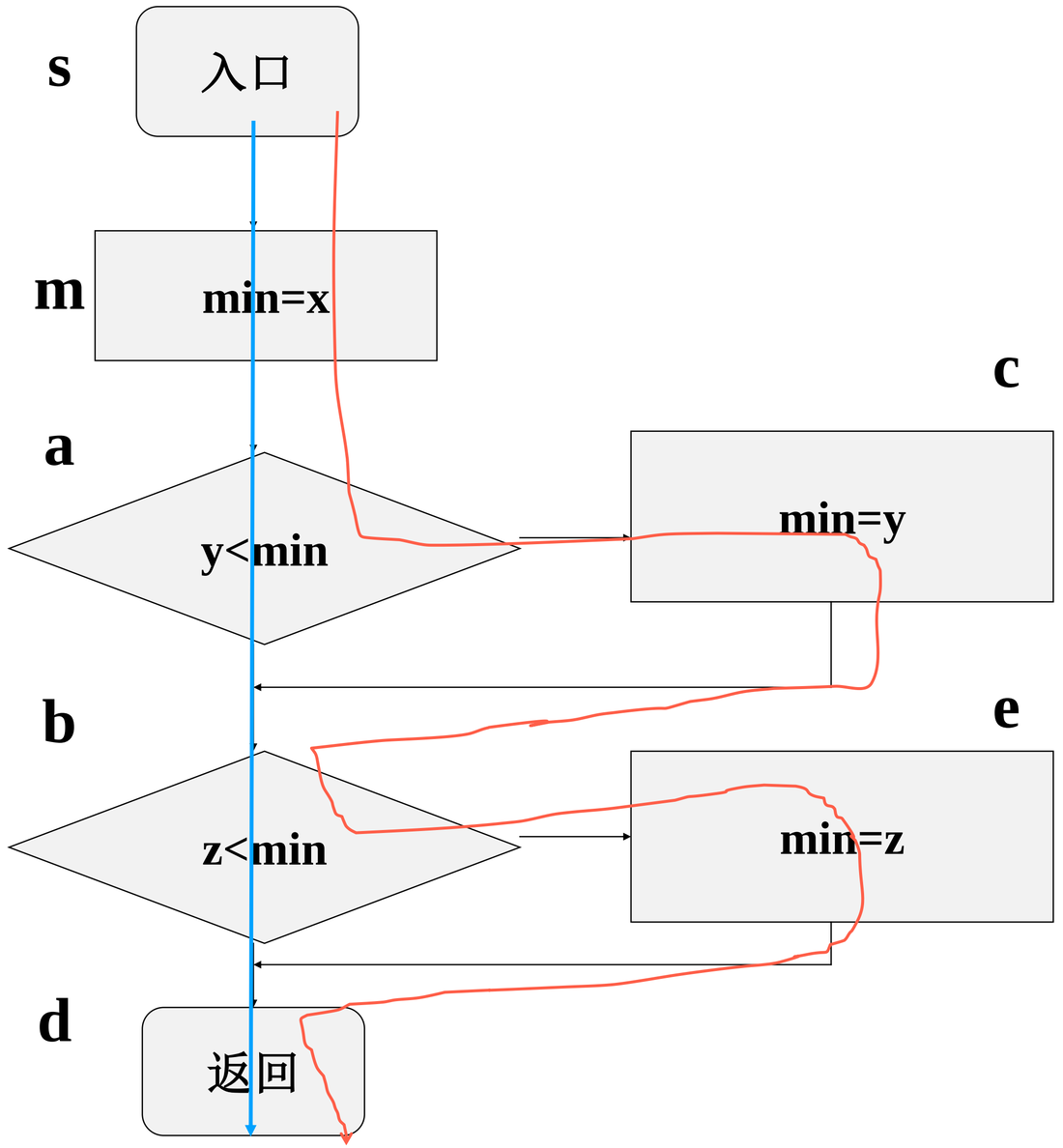
条件/判定覆盖

■ 测试用例

ID	x	y	z
1	9	8	7
2	7	8	9

■ 测试用例执行路径

- smacbed
- smabd



05 条件/判定覆盖

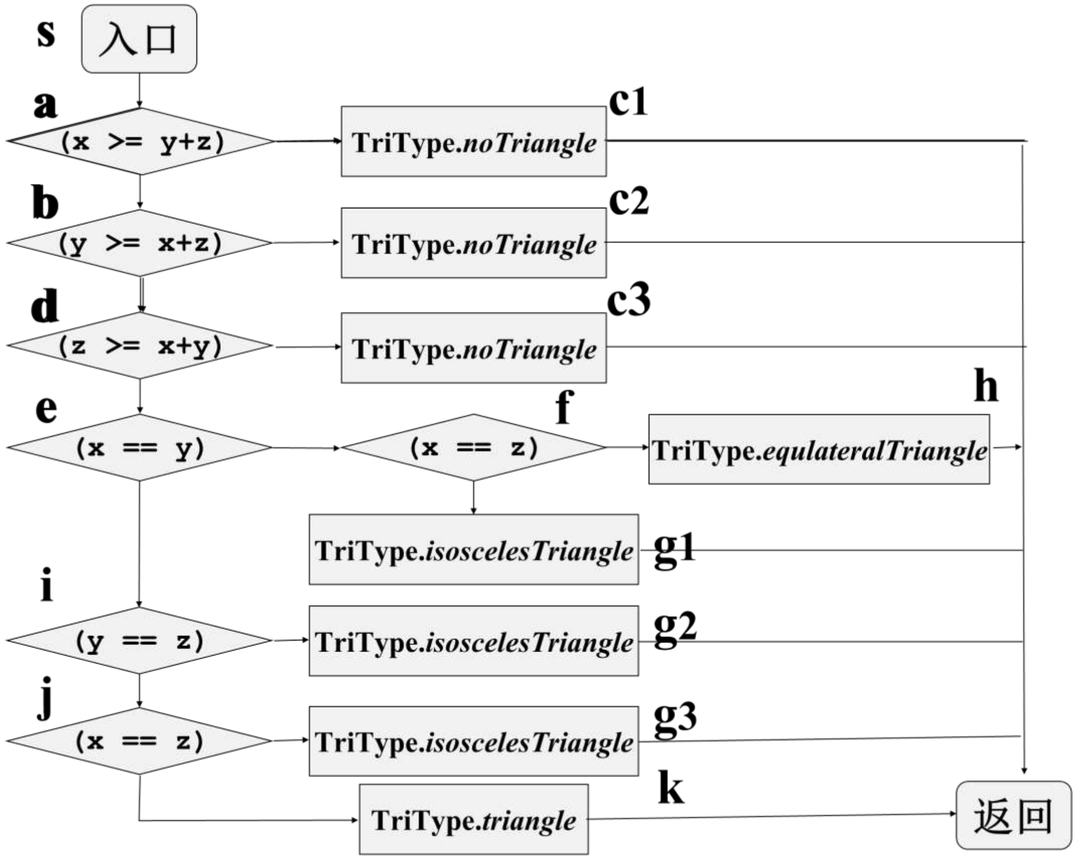
```
1  package dubo;
2  public class Triangle {
3      public TriType ReturnTriangleType(int x, int y, int z) {
4          if (x >= y + z)
5              return TriType.noTriangle;
6          if (y >= x + z)
7              return TriType.noTriangle;
8          if (z >= x + y)
9              return TriType.noTriangle;
10         if (x == y)
11             if (x == z)
12                 return TriType.equilateralTriangle;
13             else
14                 return TriType.isoscelesTriangle;
15         else if (y == z)
16             return TriType.isoscelesTriangle;
17         else if (x == z)
18             return TriType.isoscelesTriangle;
19         else
20             return TriType.triangle;
21     }
22 }
23 enum TriType {
24     equilateralTriangle, isoscelesTriangle, triangle, noTriangle
25
26 }
```

05

条件/判定覆盖

■ 测试用例

ID	x	y	z
1	9	1	1
2	1	9	1
3	1	1	9
4	1	1	1
5	2	2	3
6	3	2	2
7	2	3	2
8	2	3	4



目录

CONTENTS

06

条件组合覆盖

■ 条件覆盖

- 选取足够多的测试数据，使被测试程序中每个判定表达式中的每个条件都取到各种可能的结果。

■ 条件组合覆盖

- 选取足够多的测试数据，使得判定表达式中条件的各种可能组合都至少出现一次。

06 条件组合覆盖

```
float Compute(float A,float B,float X)
{
    if (A>1)&&(B==0) X=X/A;
    if (A==2)||(X>1) X=X+1;
    return X;
}
```

06 条件组合覆盖

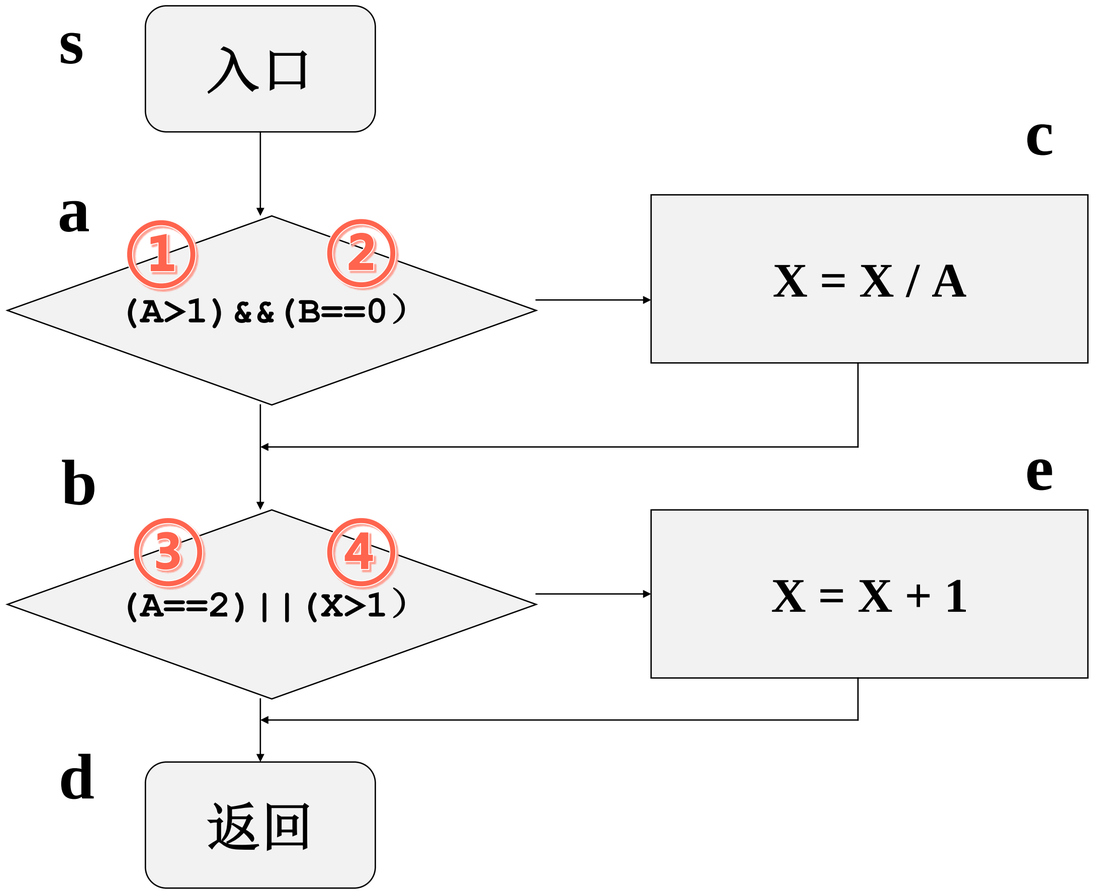
```
float Compute(float A,float B,float X)

{
      if (A>1)&&(B==0) X=X/A;

      if (A==2)||(X>1) X=X+1;

      return X;

}
```



04

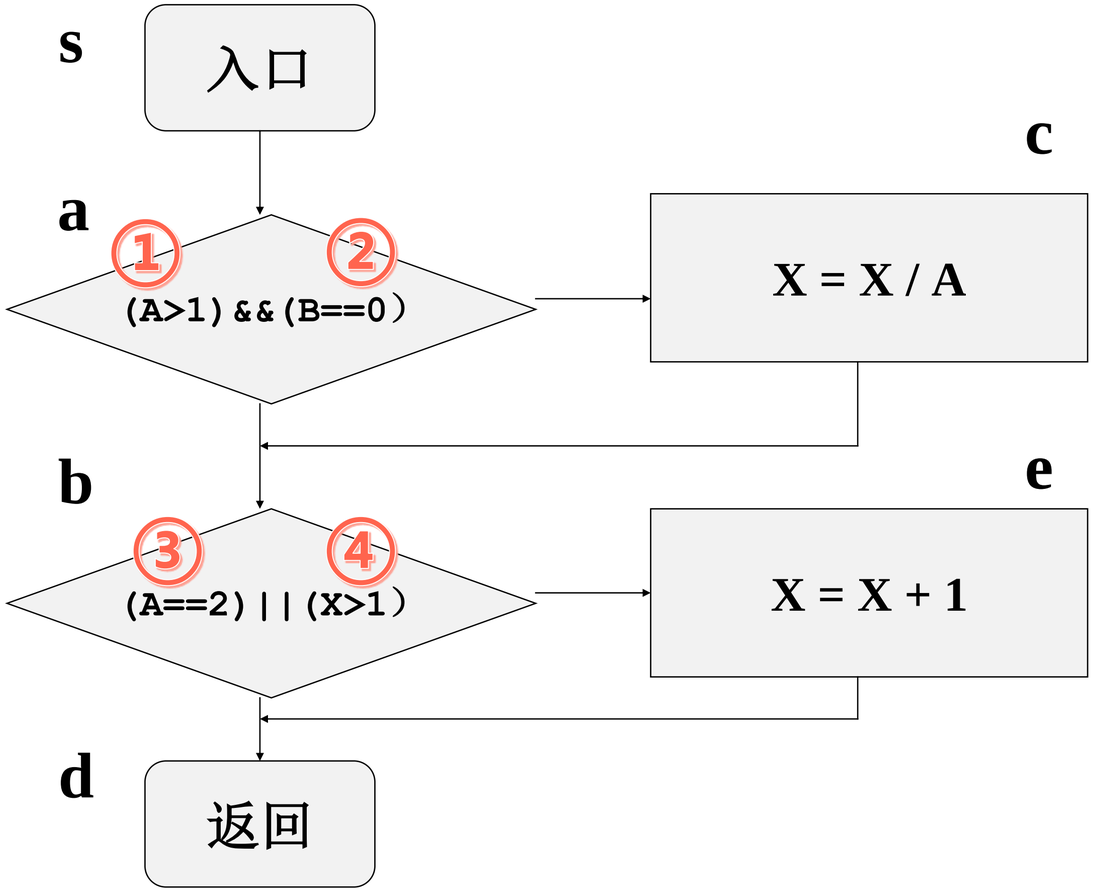
条件组合覆盖

■ 测试用例

ID	A	B	X
1	2	0	4
2	2	1	1
3	1	0	2
4	1	1	1

■ 测试用例覆盖的条件组合

- (1) A>1, B=0;
- (2) A>1, B ≠0;
- (3) A ≤1, B=0;
- (4) A ≤1, B ≠0;
- (5) A=2, X>1;
- (6) A=2, X ≤1;
- (7) A ≠ 2, X>1;
- (8) A ≠ 2, X ≤1



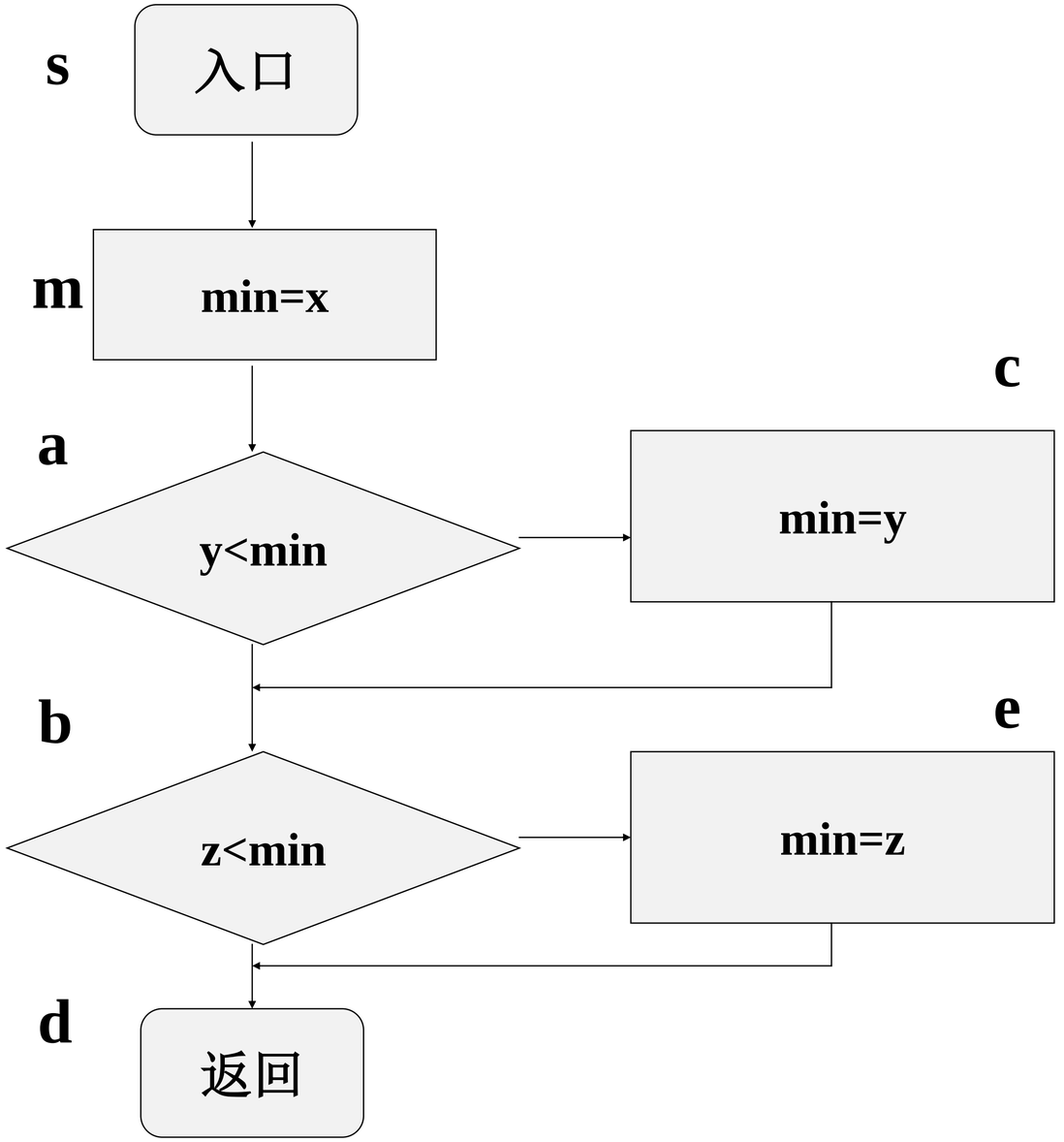
06 条件组合覆盖

```java
package dubo;

public class Comparator {

    public int min(int x, int y, int z) {
        int min = x;
        if (y < min)
            min = y;
        if (z < min)
            min = z;
        return min;
    }
}

```



06

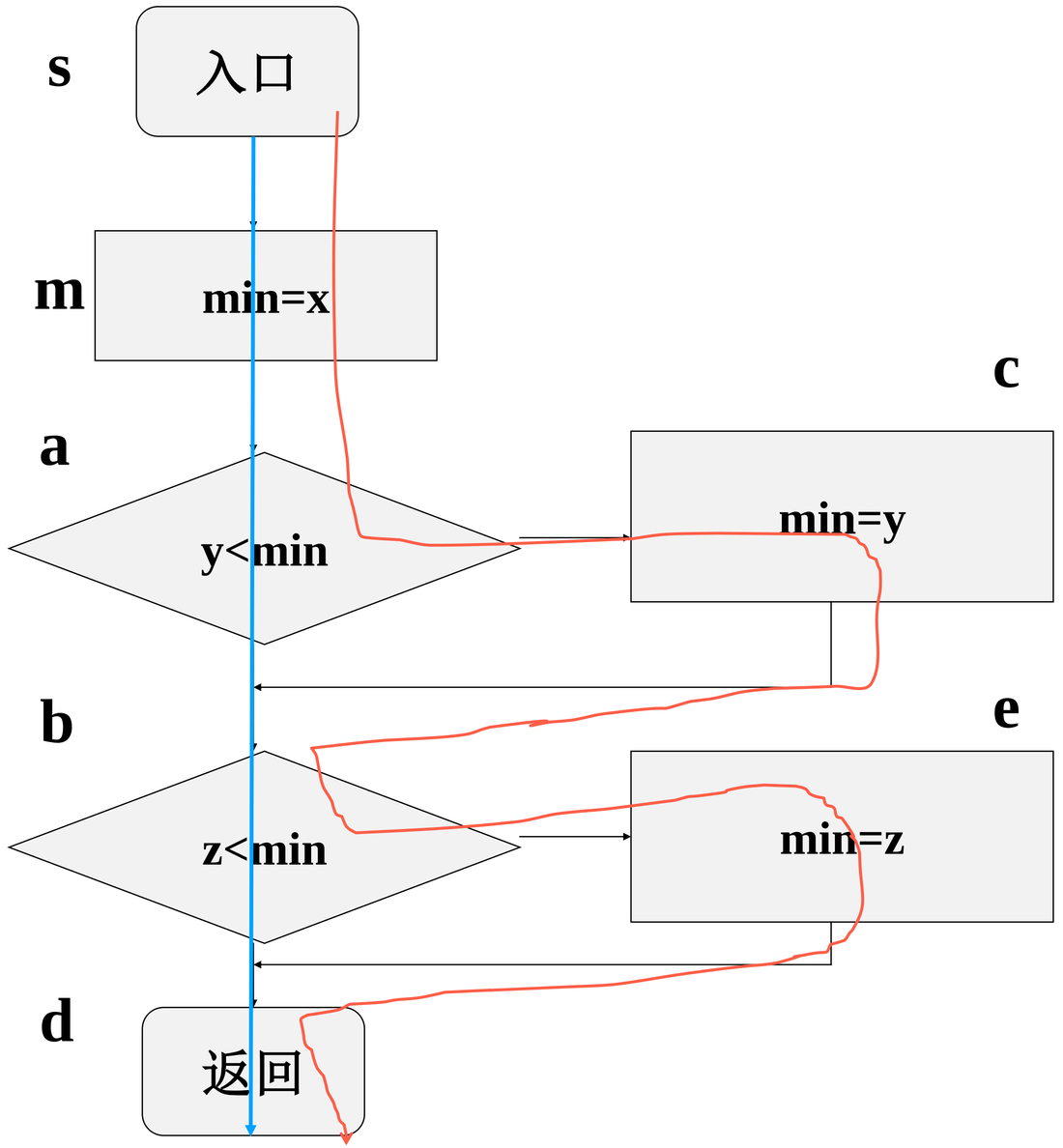
条件组合覆盖

■ 测试用例

ID	x	y	z
1	9	8	7
2	7	8	9

■ 测试用例执行路径

- smacbed
- smabd



06 条件组合定覆盖

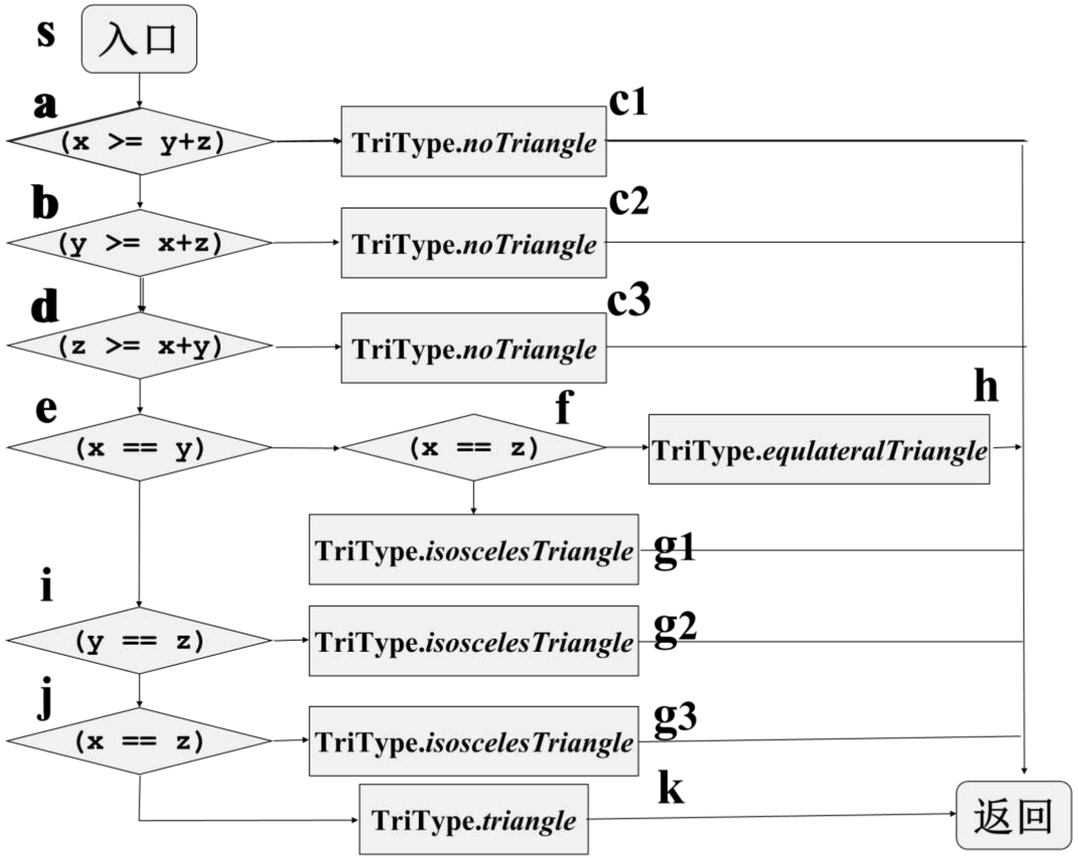
```
1  package dubo;
2  public class Triangle {
3      public TriType ReturnTriangleType(int x, int y, int z) {
4          if (x >= y + z)
5              return TriType.noTriangle;
6          if (y >= x + z)
7              return TriType.noTriangle;
8          if (z >= x + y)
9              return TriType.noTriangle;
10         if (x == y)
11             if (x == z)
12                 return TriType.equilateralTriangle;
13             else
14                 return TriType.isoscelesTriangle;
15         else if (y == z)
16             return TriType.isoscelesTriangle;
17         else if (x == z)
18             return TriType.isoscelesTriangle;
19         else
20             return TriType.triangle;
21     }
22 }
23 enum TriType {
24     equilateralTriangle, isoscelesTriangle, triangle, noTriangle
25
26 }
```

06

条件组合覆盖

■ 测试用例

ID	x	y	z
1	9	1	1
2	1	9	1
3	1	1	9
4	1	1	1
5	2	2	3
6	3	2	2
7	2	3	2
8	2	3	4



06

条件组合覆盖

谁更严格更全面呢?

■ 条件组合覆盖

- 选取足够多的测试数据，使得判定表达式中条件的各种可能组合都至少出现一次。

■ 条件/判定覆盖

- 选取足够多的测试数据，使得判定表达式中的每个条件都取到各种可能的结果，**而且**每个判定表达式也都取到各种可能的结果。
- 条件/判定覆盖==条件覆盖+判定覆盖

目录

CONTENTS

07

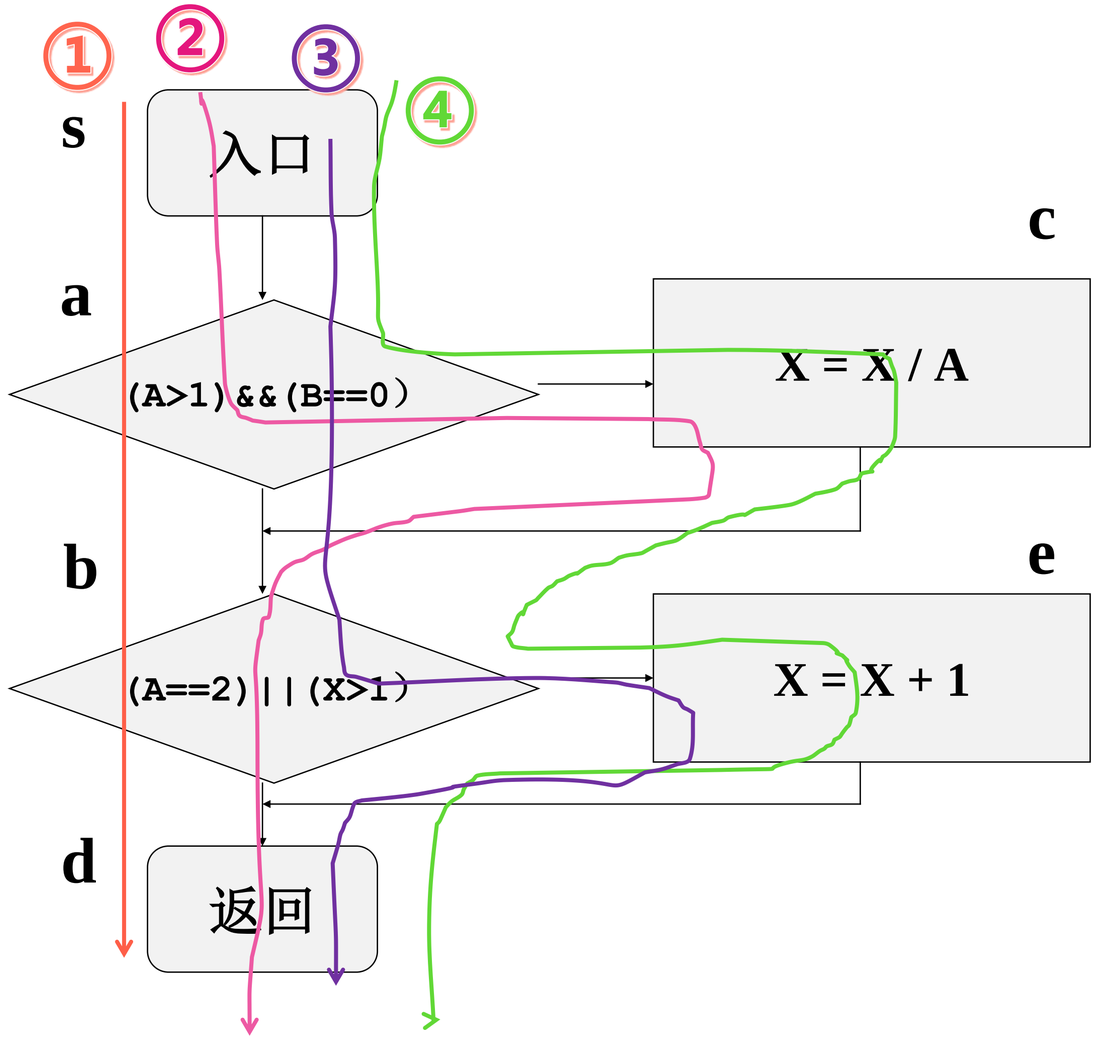
路径覆盖

■ 路径覆盖

- 选取足够多的测试数据，使得程序的每条可能路径都至少执行一次（若程序图中有环，则每个环至少经过一次）。

07 路径覆盖

```
float Compute(float A,float B,float X)
{
    if (A>1)&&(B==0) X=X/A;
    if (A==2)||(X>1) X=X+1;
    return X;
}
```



07

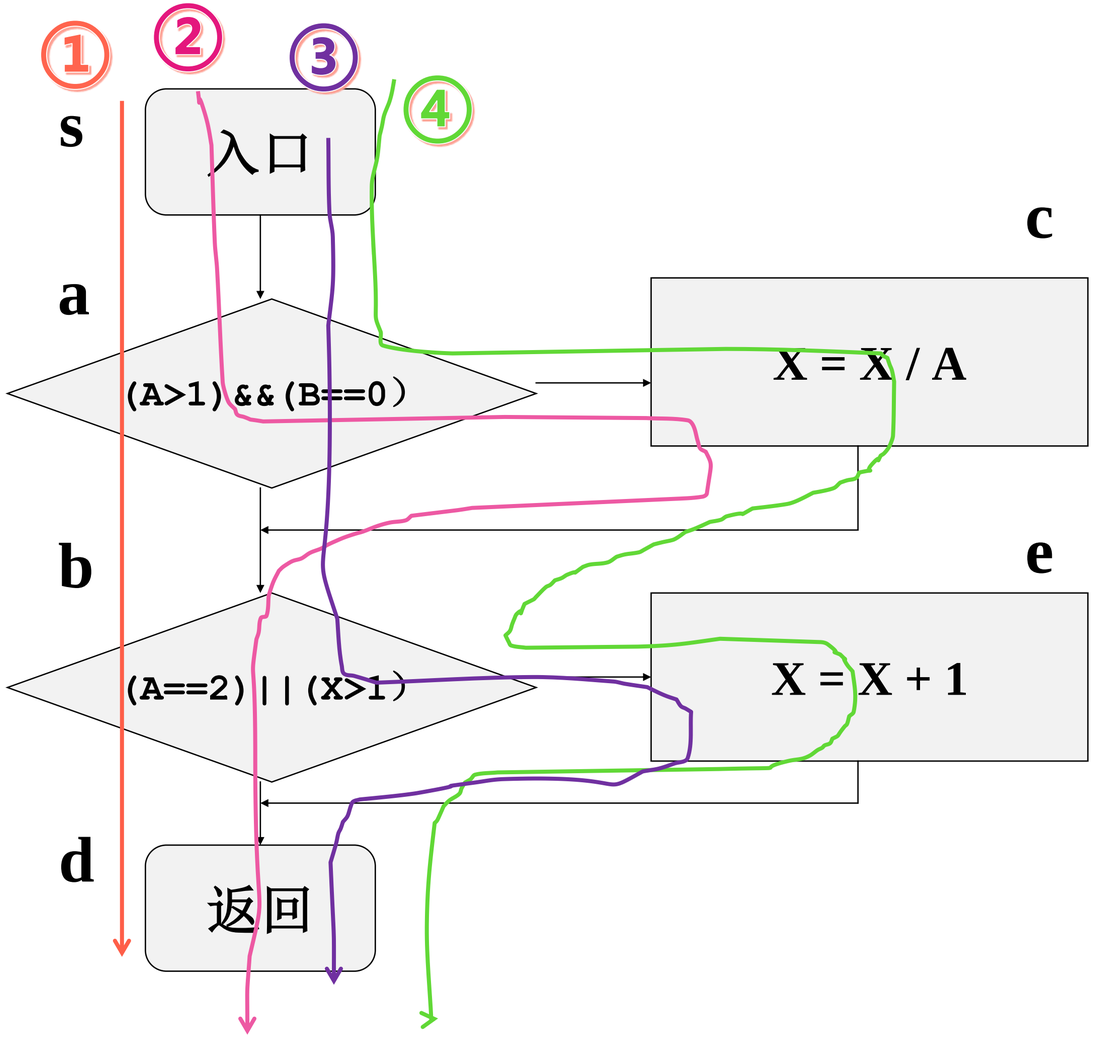
路径覆盖

■ 测试用例

ID	A	B	X
1	0	0	0
2	3	0	1
3	2	1	2
4	2	0	6

■ 测试用例执行路径

- sabd
- sacbd
- sabed
- sacbed



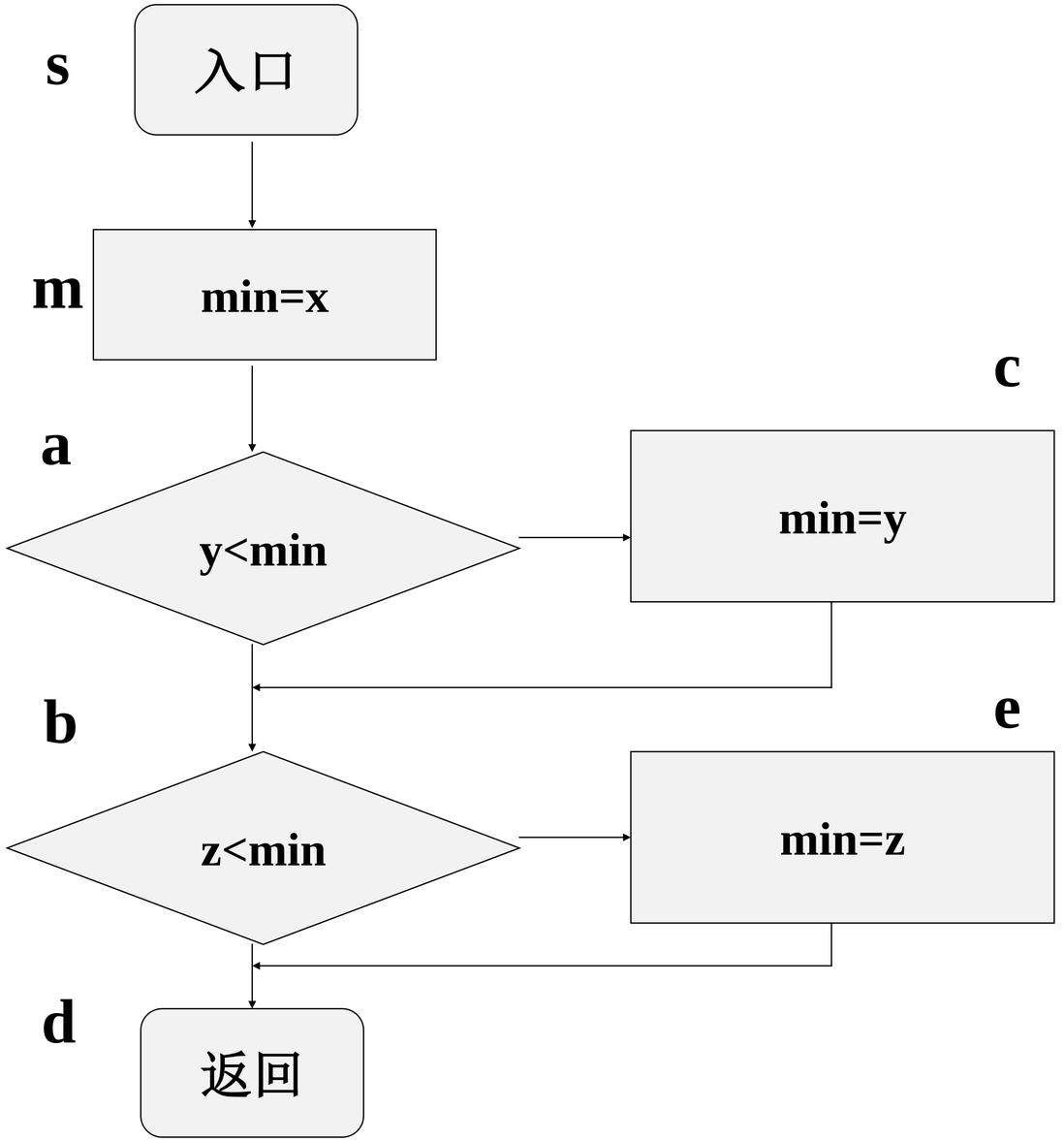
07 路径覆盖

```java
package dubo;

public class Comparator {

    public int min(int x, int y, int z) {
        int min = x;
        if (y < min)
            min = y;
        if (z < min)
            min = z;
        return min;
    }
}

```



07

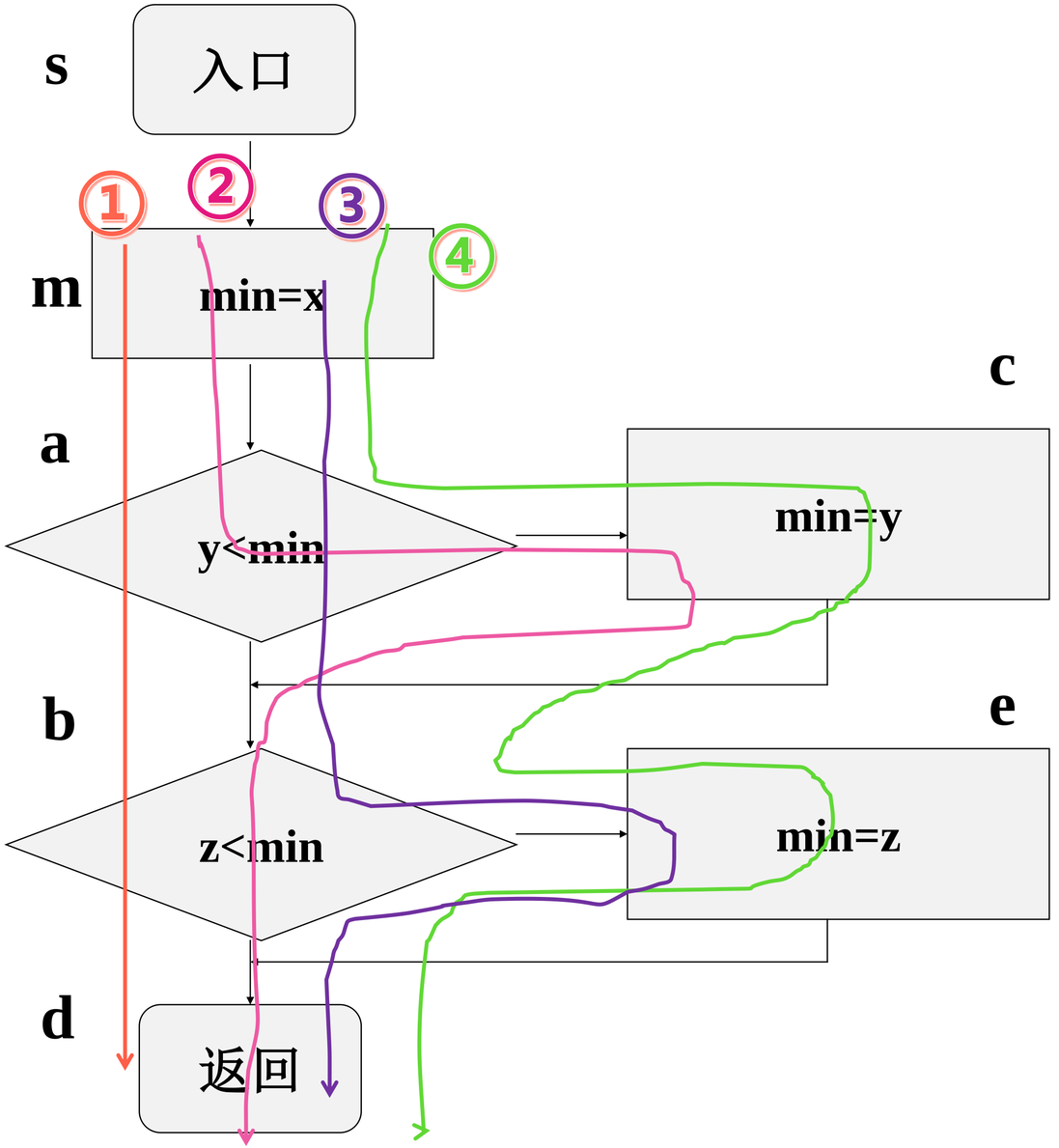
路径覆盖

■ 测试用例

ID	x	y	z
1	7	8	9
2	6	5	7
3	6	7	5
4	9	8	6

■ 测试用例执行路径

- smabd
- Smacbd
- Smabed
- smacbed



07 路径覆盖

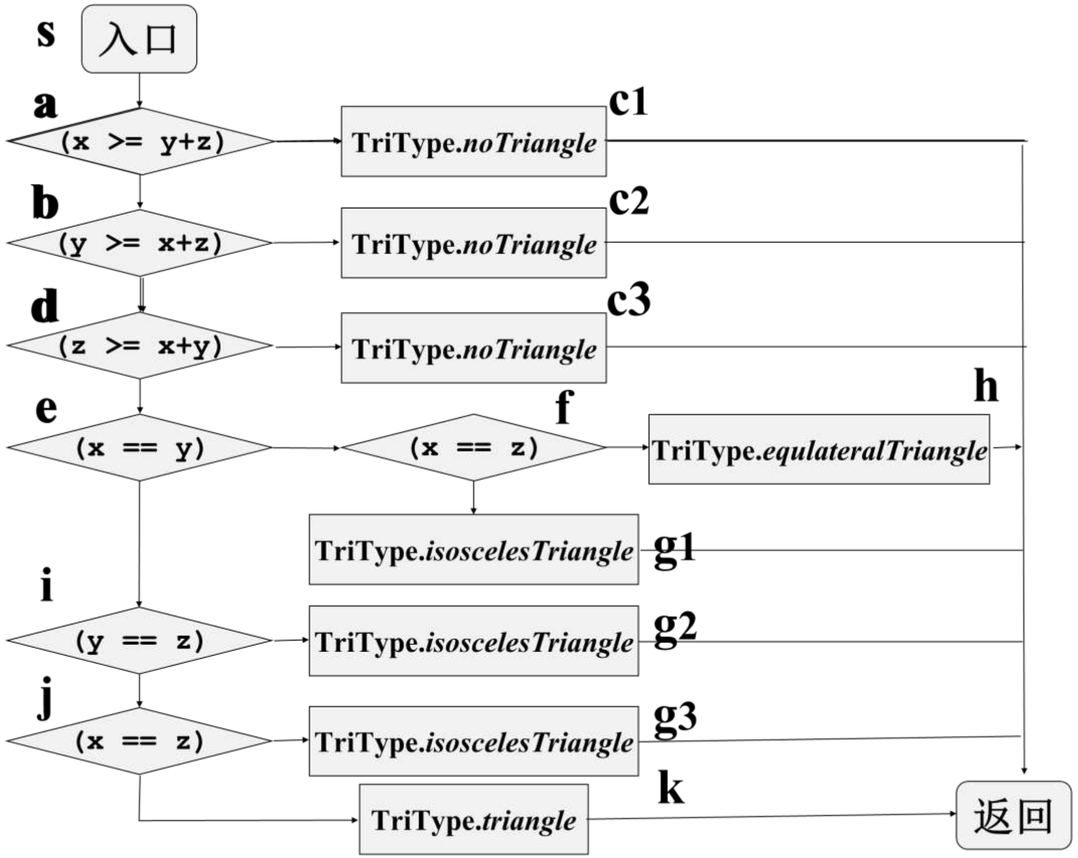
```
1  package dubo;
2  public class Triangle {
3      public TriType ReturnTriangleType(int x, int y, int z) {
4          if (x >= y + z)
5              return TriType.noTriangle;
6          if (y >= x + z)
7              return TriType.noTriangle;
8          if (z >= x + y)
9              return TriType.noTriangle;
10         if (x == y)
11             if (x == z)
12                 return TriType.equilateralTriangle;
13             else
14                 return TriType.isoscelesTriangle;
15         else if (y == z)
16             return TriType.isoscelesTriangle;
17         else if (x == z)
18             return TriType.isoscelesTriangle;
19         else
20             return TriType.triangle;
21     }
22 }
23 enum TriType {
24     equilateralTriangle, isoscelesTriangle, triangle, noTriangle
25
26 }
```

06

路径覆盖

■ 测试用例

ID	x	y	z
1	9	1	1
2	1	9	1
3	1	1	9
4	1	1	1
5	2	2	3
6	3	2	2
7	2	3	2
8	2	3	4



目录

CONTENTS

08 小结

白盒测试利用程序内部的逻辑结构及有关信息，设计或选择测试用例

白盒测试以代码覆盖率为主要目标

可以采用不同的覆盖率计算标准，包括语句、分支、条件、路径等

谢谢