



**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

*Nhóm chuyên môn Nhập môn Công nghệ phần mềm*

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

**Phương pháp kiểm thử hộp trắng (phần 1)**



**1. Khái niệm kiểm thử hộp trắng**

**2. Kiểm thử luồng điều khiển**

*Sau bài học này, người học có thể:*

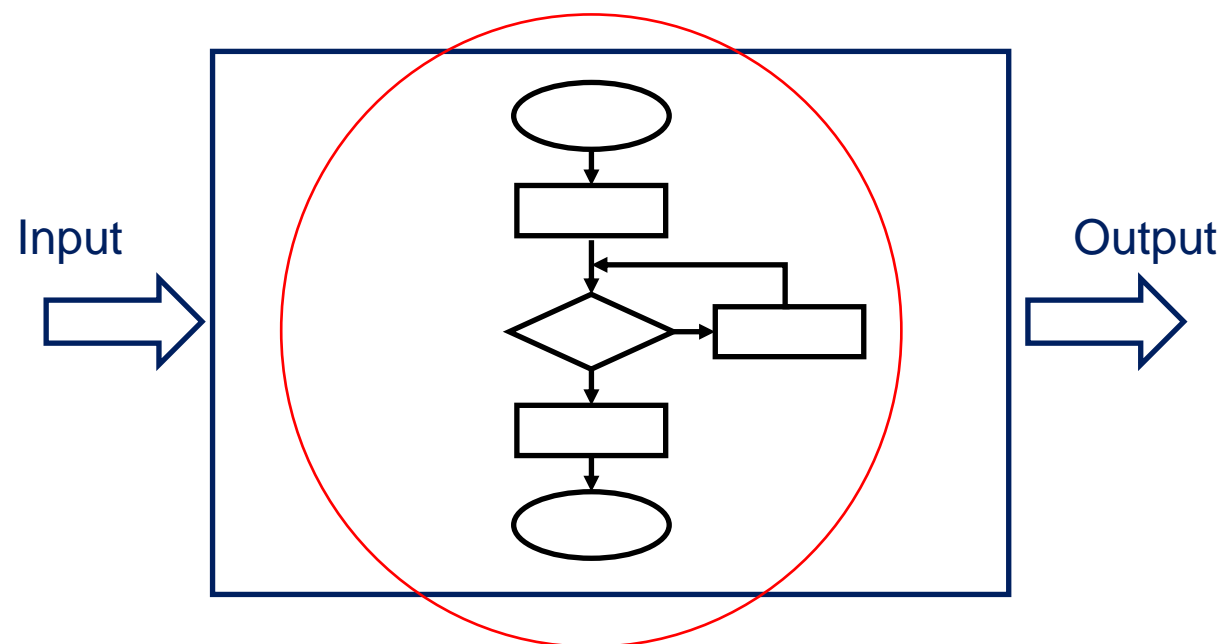
1. Hiểu về kỹ thuật kiểm thử hộp trắng
2. Nắm được phương pháp kiểm thử luồng điều khiển

## 1. Khái niệm kiểm thử hộp trắng

## 2. Kiểm thử luồng điều khiển

# 1. KHÁI NIỆM KIỂM THỬ HỘP TRẮNG

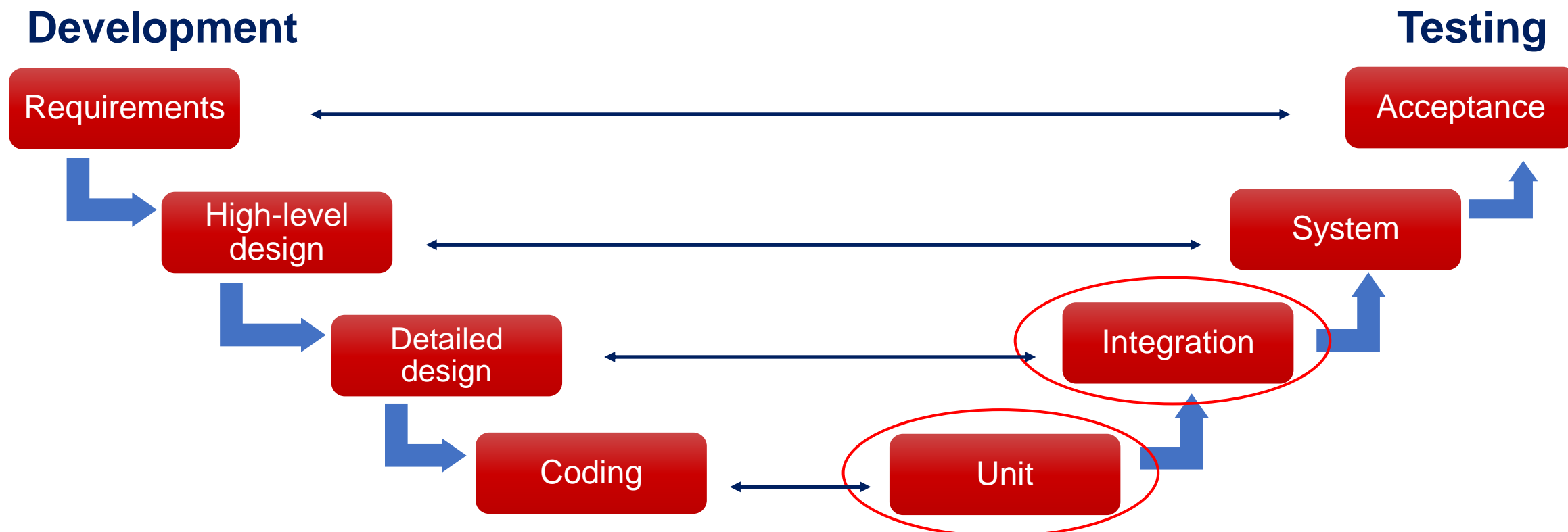
- Kiểm thử hộp trắng là chiến lược kiểm thử dựa trên **cấu trúc chức năng** (structural testing) nhằm **kiểm tra về mặt thuật toán và cấu trúc bên trong** của sản phẩm phần mềm.



Hình 1.1: Kiểm thử hộp trắng

# 1. KHÁI NIỆM KIỂM THỬ HỘP TRẮNG

- Được thực hiện bởi lập trình viên và các kiểm thử viên;
- Thực hiện chủ yếu ở giai đoạn kiểm thử đơn vị và kiểm thử tích hợp.



Hình 1.2: Mô hình chữ V và kiểm thử hộp trắng

# 1. KHÁI NIỆM KIỂM THỬ HỘP TRẮNG

---



## Các phương pháp thiết kế ca kiểm thử

- ***Kiểm thử luồng điều khiển***

- Bao phủ tất cả đường dẫn
- Bao phủ lệnh
- Bao phủ nhánh

- ***Kiểm thử luồng dữ liệu***

- Bao phủ tất cả các điểm định nghĩa của biến
- Bao phủ tất cả các điểm sử dụng của biến
- ...

1. Khái niệm kiểm thử hộp trắng

**2. Kiểm thử luồng điều khiển**



## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

---

### Đồ thị luồng điều khiển (Control Flow Graph - CFG)

- Đồ thị luồng điều khiển thể hiện tập hợp tất cả các đường dẫn từ điểm bắt đầu tới điểm kết thúc của chương trình được biểu diễn bằng đồ thị;
- Biểu diễn cấu trúc của một đơn vị chương trình.

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

---

### Đồ thị luồng điều khiển (CFG)

- Một đường dẫn (path) của chương trình
  - Là một chuỗi các câu lệnh (statement) từ điểm bắt đầu cho đến điểm kết thúc của chương trình
  - Với mỗi một dữ liệu đầu vào (input data), chương trình có thể thực thi theo các đường dẫn khác nhau (execution path)
- Mục đích của việc thiết kế các ca kiểm thử
  - Tìm ra những đường dẫn (path) có lỗi hoặc tiềm ẩn nhiều lỗi nhất có thể

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

---



### Các tiêu chí lựa chọn đường dẫn chương trình

- *Bao phủ toàn bộ các đường dẫn (vết càn)*
  - Nếu chương trình có 10 lệnh rẽ nhánh thì số đường dẫn tối đa là  $2^{10} = 1024$  đường dẫn, nên có 1024 test case.
- *Bao phủ lệnh*
  - Đảm bảo mọi câu lệnh trong chương trình đều được kiểm tra
- *Bao phủ nhánh*
  - Đảm bảo mọi nhánh rẽ tương ứng với giá trị true/false ở câu lệnh điều kiện trong chương trình đều được kiểm thử.

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

---

### Ví dụ

- Cho đoạn chương trình sau

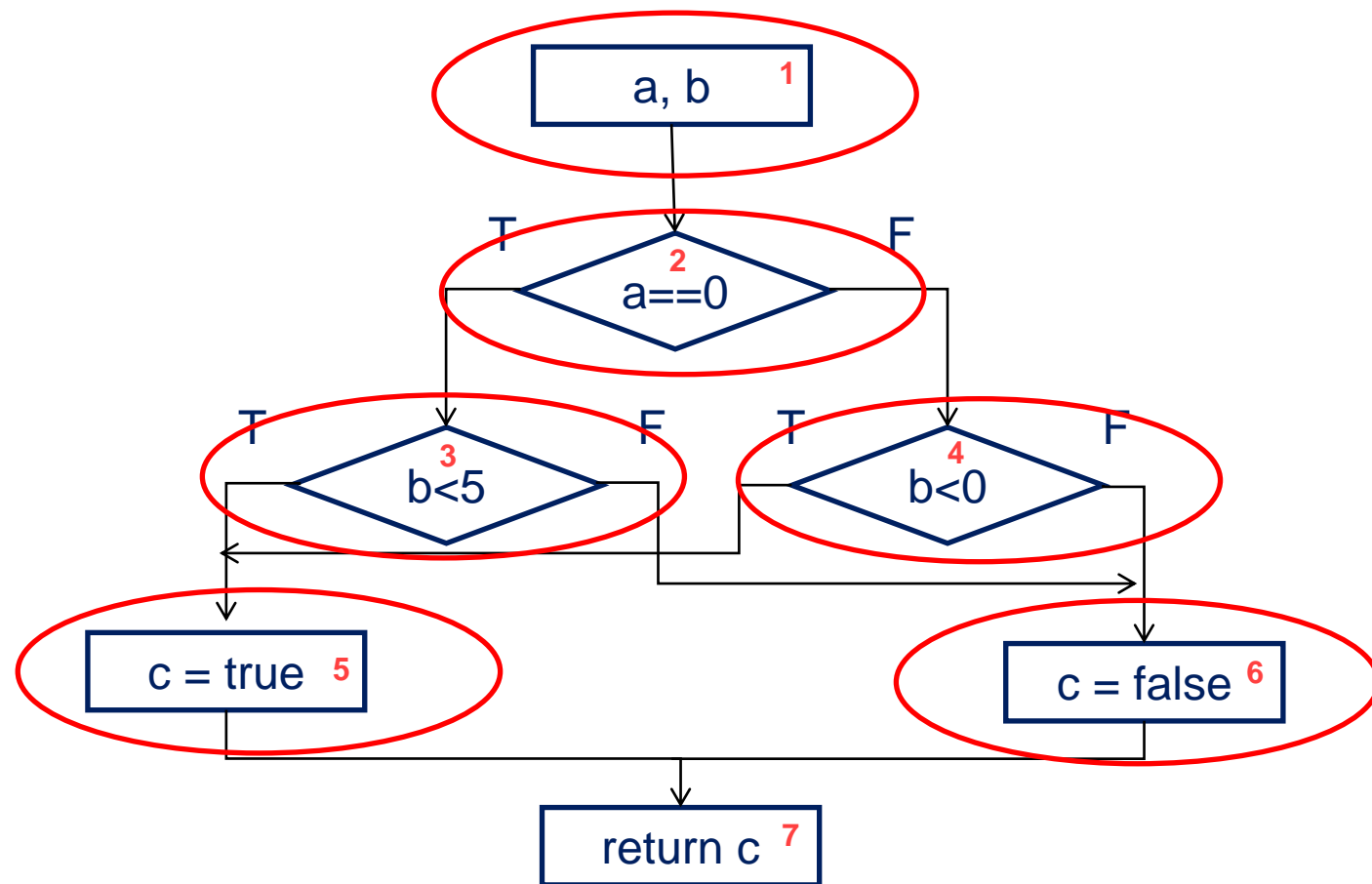
```
bool foo(int a, int b) {  
    bool c;  
    if (a==0)  
        c = b < 5;  
    else  
        c = b < 0;  
    return c;  
}
```

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Ví dụ

- Vẽ sơ đồ luồng điều khiển

```
bool foo(int a, int b) {  
    bool c;  
    if (a==0)  
        c = b < 5;  
    else  
        c = b < 0;  
    return c;  
}
```



Hình 2.1: Sơ đồ luồng điều khiển

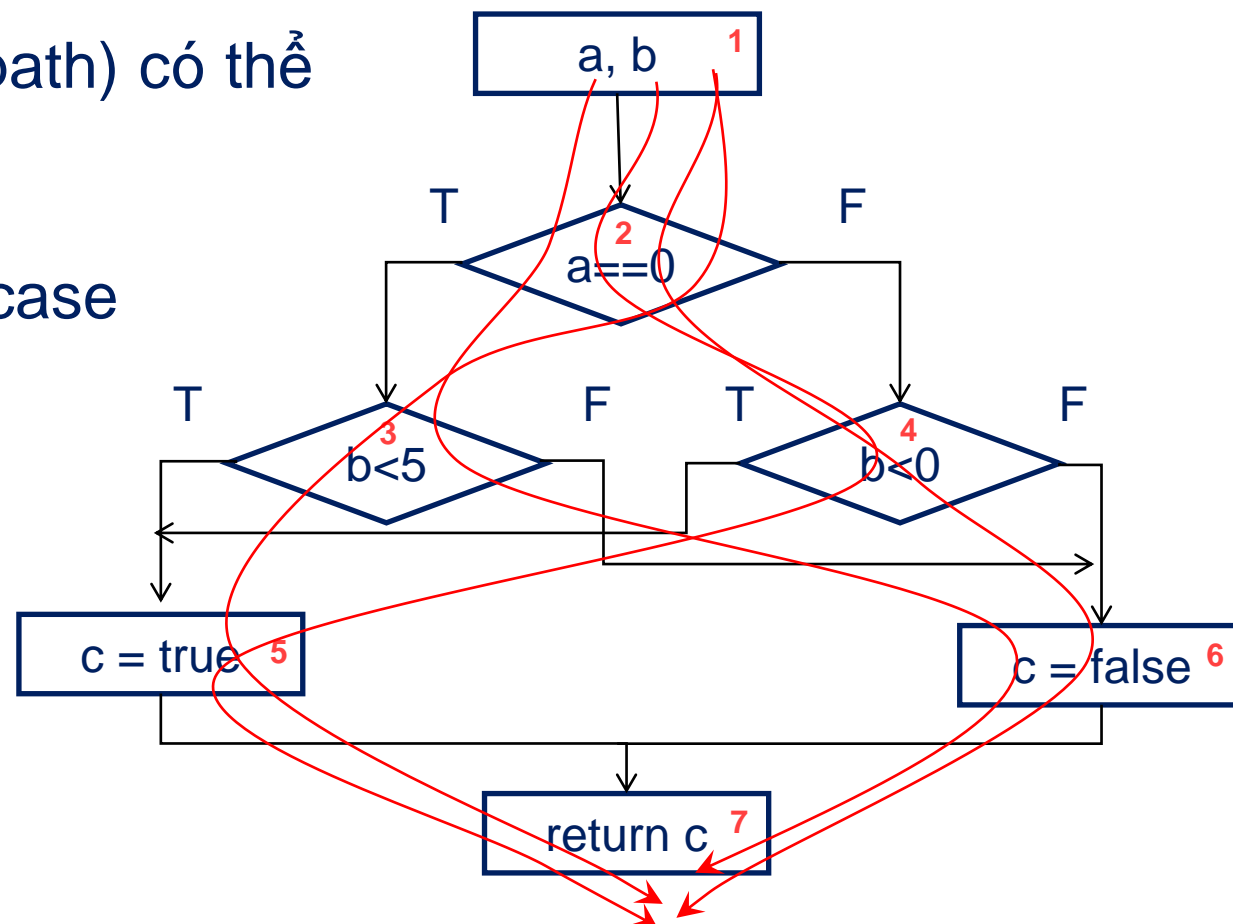
## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Tiêu chí phủ đường dẫn

- Lựa chọn tất cả các đường dẫn (path) có thể có trong chương trình
- Mỗi path sẽ tương ứng với 1 test case

- Xác định các đường dẫn

- 1) 1-2-3-5-7
- 2) 1-2-3-6-7
- 3) 1-2-4-5-7
- 4) 1-2-4-6-7

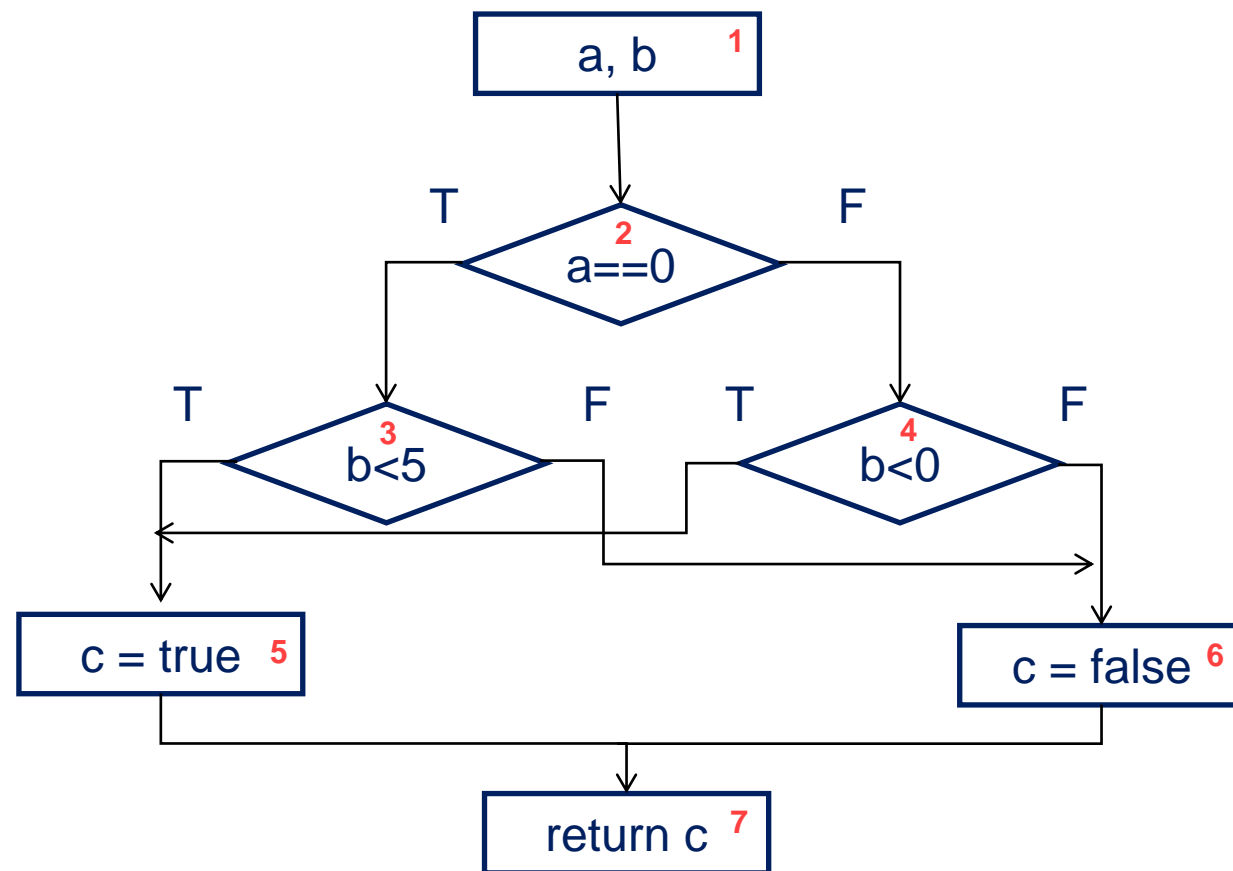


Hình 2.1: Sơ đồ luồng điều khiển

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Tiêu chí phủ đường dẫn

	a==0	b<5	b<0	return
1	T	T	T	T
2	T	T	F	T
<u>3</u>	<u>T</u>	<u>F</u>	<u>T</u>	-
4	T	F	F	F
5	F	T	T	T
<u>6</u>	<u>F</u>	<u>F</u>	<u>T</u>	-
7	F	T	F	F
8	F	F	F	F



Hình 2.1: Sơ đồ luồng điều khiển

# 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

## Tiêu chí phủ đường dẫn

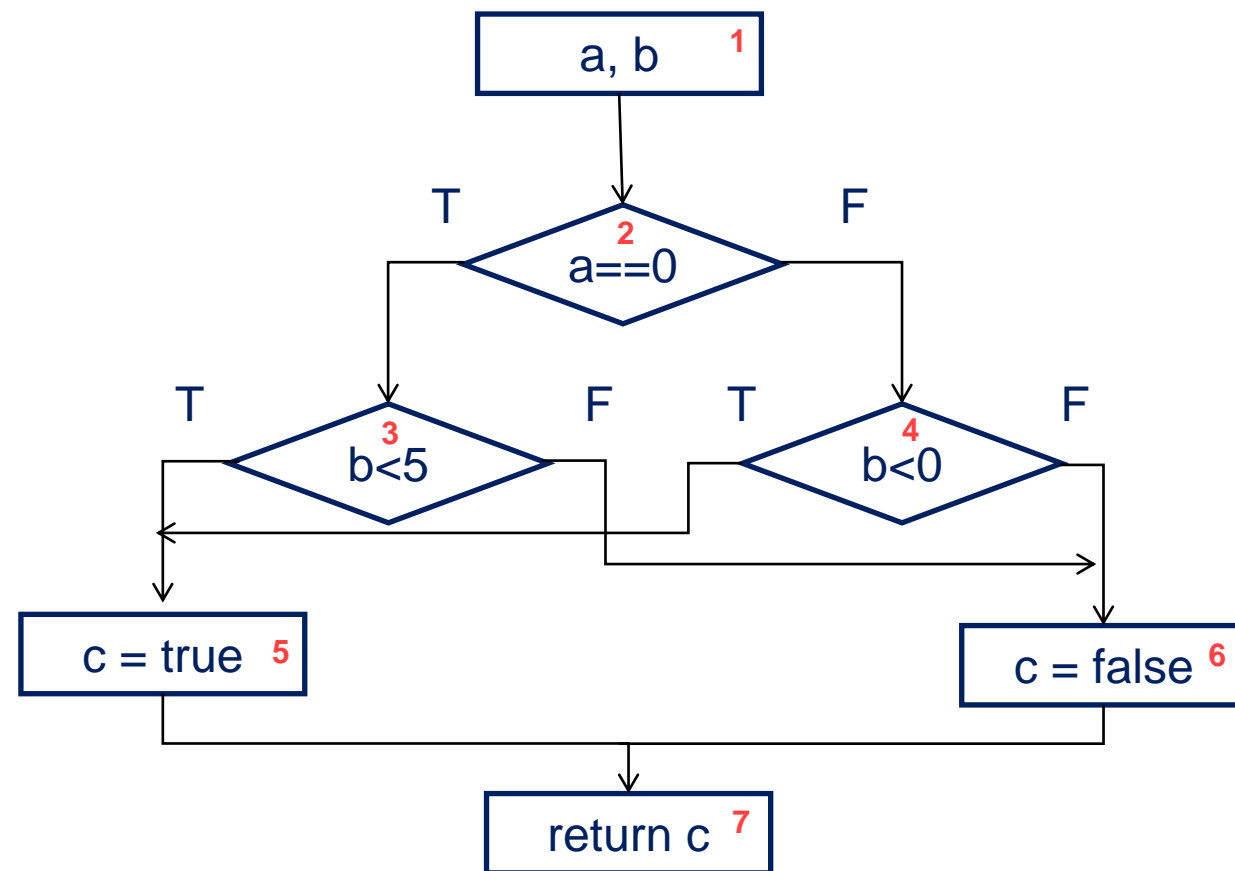
		a==0	b<5	b<0	return
1)	1	T	T	T	T
	2	T	T	F	T
2)	3	F	F	F	-
	4	T	F	F	F
3)	5	F	T	T	T
	6	F	F	F	-
4)	7	F	T	F	F
	8	F	F	F	F

1)  $\langle T, T, * \rangle$  1-2(T)-3(T)-5-7

2)  $\langle T, F, F \rangle$  1-2(T)-3(F)-6-7

3)  $\langle F, T, T \rangle$  1-2(F)-4(T)-5-7

4)  $\langle F, *, F \rangle$  1-2(F)-4(F)-6-7



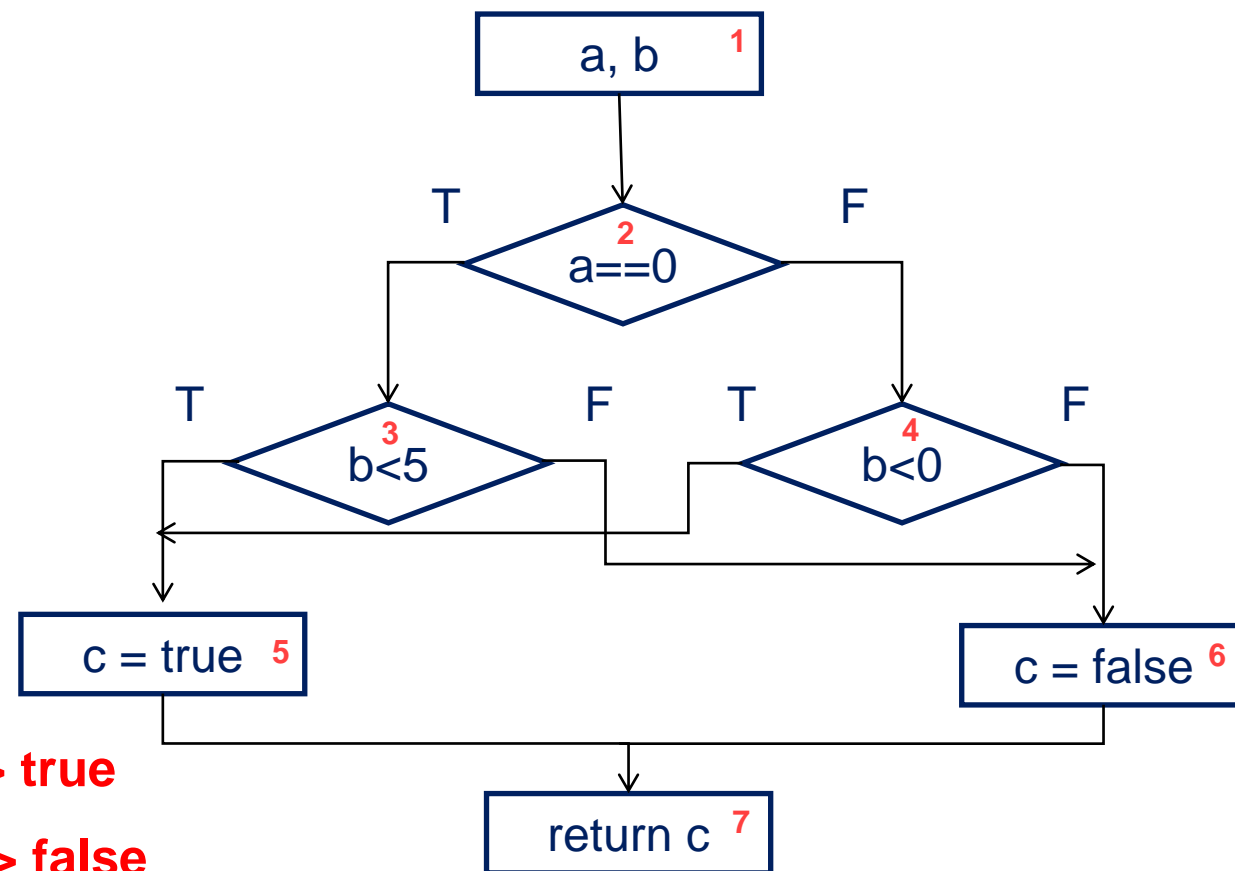
Hình 2.1: Sơ đồ luồng điều khiển



## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Tiêu chí phủ đường dẫn

	a==0	b<5	b<0	return
1)	1	T	T	T
	2	T	F	T
	3	F	F	-
2)	4	T	F	F
3)	5	F	T	T
	6	F	F	-
4)	7	F	T	F
	8	F	F	F



- 1)  $\langle T, T, * \rangle$  1-2(T)-3(T)-5-7  $\Rightarrow$  TC: foo(0,3)  $\Rightarrow$  true
- 2)  $\langle T, F, F \rangle$  1-2(T)-3(F)-6-7  $\Rightarrow$  TC: foo(0,6)  $\Rightarrow$  false
- 3)  $\langle F, T, T \rangle$  1-2(F)-4(T)-5-7  $\Rightarrow$  TC: foo(1,-2)  $\Rightarrow$  true
- 4)  $\langle F, *, F \rangle$  1-2(F)-4(F)-6-7  $\Rightarrow$  TC: foo(1,3)  $\Rightarrow$  false

Hình 2.1: Sơ đồ luồng điều khiển

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

---



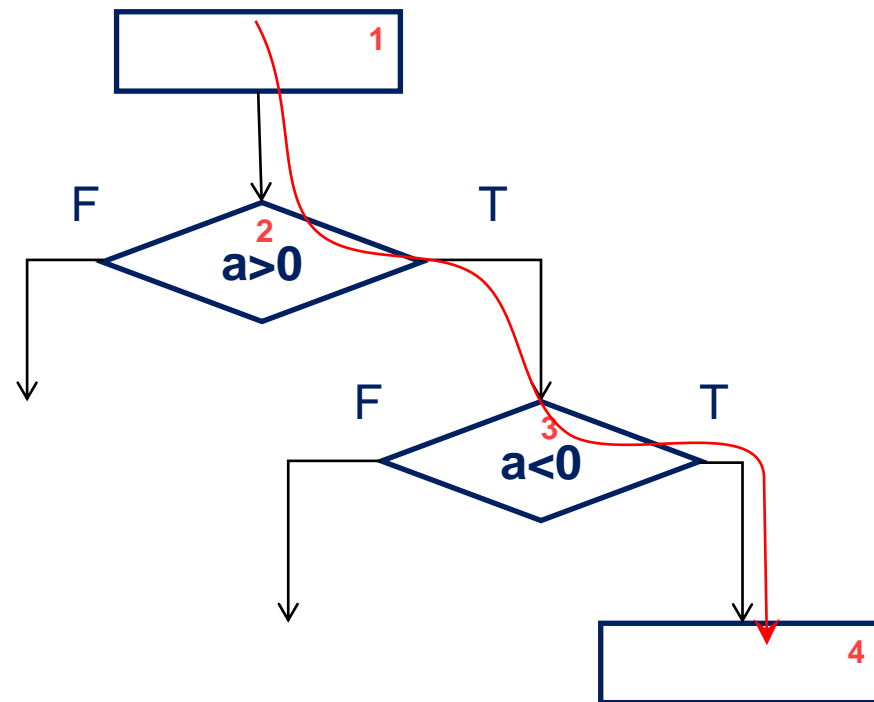
### Nhận xét (1)

- Với bài toán kiểm thử đường dẫn ta cần xây dựng đồ thị luồng điều khiển
- Cần xác định các đường dẫn và các giá trị cần thiết của biến/tham số để thiết kế các test case

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Nhận xét (2)

- Có thể tồn tại một số đường dẫn không thể thiết kế được test case ví dụ không thể thực thi đường dẫn đi qua các node 1, 2, 3, 4 như ở hình bên

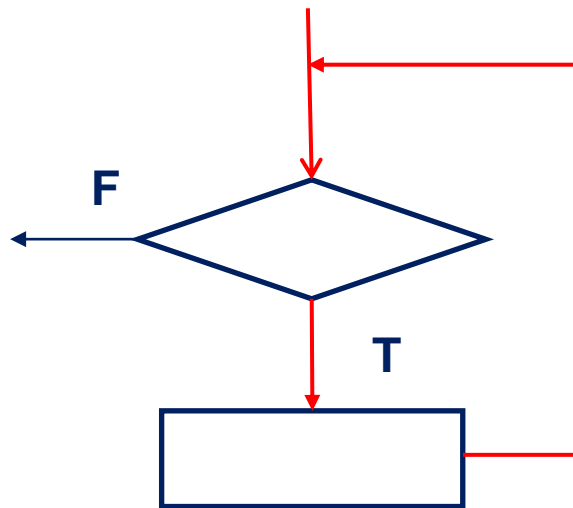


Hình 2.2: Đường đi không hợp lệ

## 2. KIỂM THỬ LUỒNG ĐIỀU KHIỂN

### Nhận xét (3)

- Trong trường hợp chương trình có câu lệnh lặp (đặc biệt với câu lệnh lặp không xác định) thì số lượng test case sẽ rất lớn (hoặc không thể xác định được).



1. Bài học đã cung cấp cho người học **kỹ thuật kiểm thử hộp trắng, phương pháp kiểm thử luồng điều khiển** (tiêu chí phủ đường dẫn)
2. Tiếp sau bài này, **người học có thể tự tìm hiểu thêm** về tiêu chí bao phủ lệnh và bao phủ nhánh

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

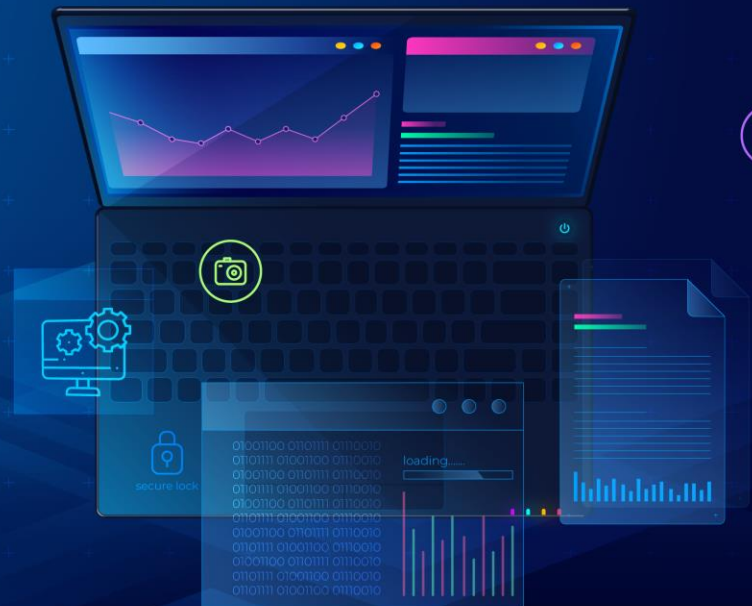
## Phương pháp kiểm thử hộp trắng (phần 1)

Biên soạn:

TS. Trần Nhật Hóa

Trình bày:

TS. Trần Nhật Hóa



# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

*Bài học tiếp theo:*

## **Phương pháp kiểm thử hộp trắng (phần 2)**

**Tài liệu tham khảo:**

- [1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016.
- [2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.
- [3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.
- [4] Shari Lawrence Pleege, Joanne M. Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009