



# KIỂM THỬ PHẦN MỀM

(Software Testing)



GV: ThS. Nguyễn Thị Thanh Trúc  
Khoa: Công nghệ Phần mềm  
Email: tructt@uit.edu.vn

# Bài 5: Các kỹ thuật kiểm thử



- **Test tĩnh (Static Verification)**
- **Test động (Dynamic Testing)**
- **5.1 Các kỹ thuật kiểm thử hộp đen**
- **5.2 Các kỹ thuật kiểm thử hộp trắng**

# Các kỹ thuật kiểm thử



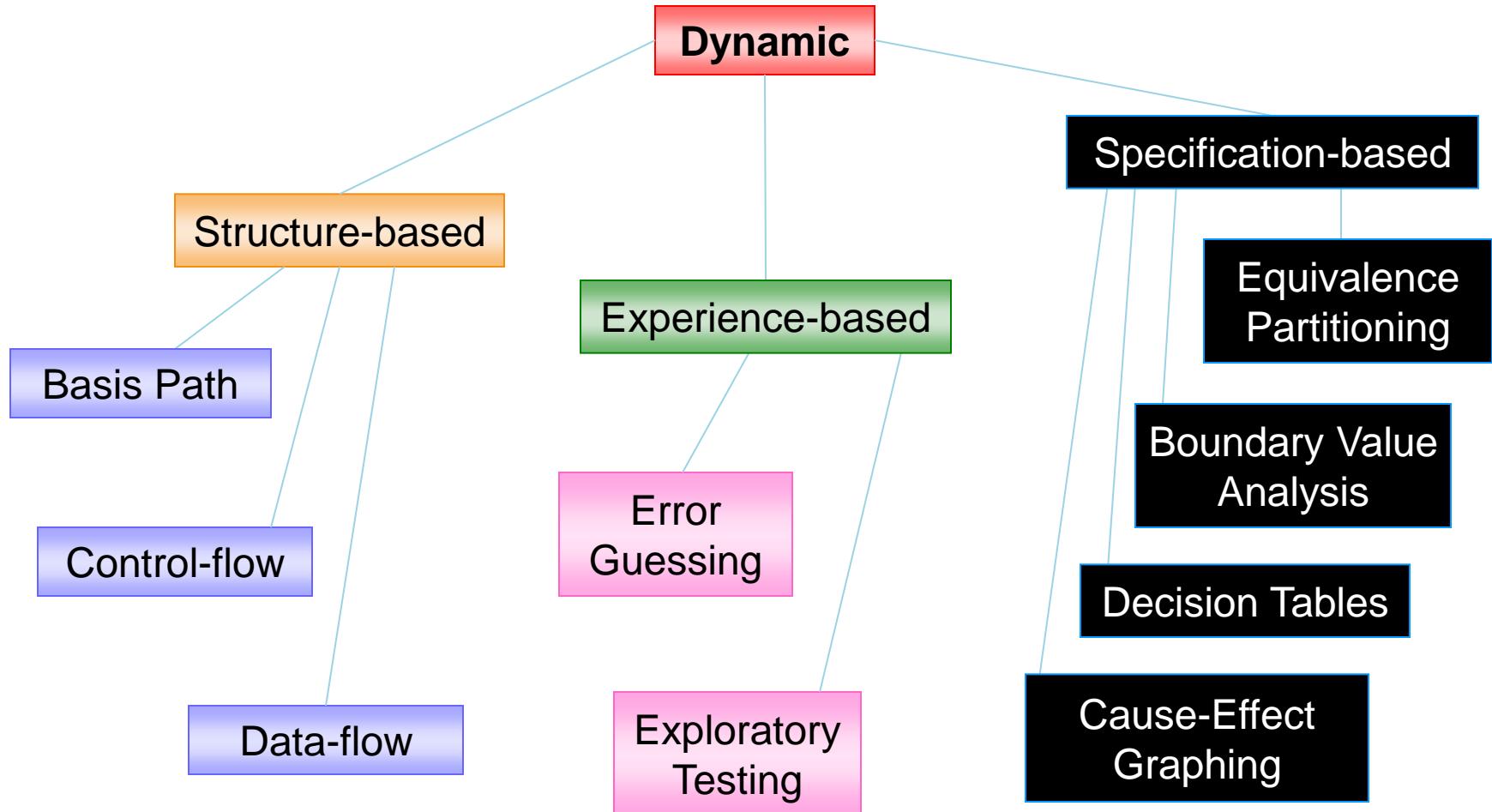
- **Test tĩnh (Static Verification)**

- Thực hiện kiểm chứng mà **không cần thực thi chương trình**
- Kiểm tra tính đúng đắn của các tài liệu có liên quan được tạo ra trong quá trình xây dựng ứng dụng
- Đạt được sự nhất quán và hiểu rõ hơn về hệ thống
- Giảm thời gian lập trình, thời gian và chi phí test,...

- **Test động (Dynamic Testing)**

- Thực hiện kiểm thử dựa trên việc thực thi chương trình

# Dynamic Testing - Kiểm thử động



# Các phương pháp kiểm thử (1)

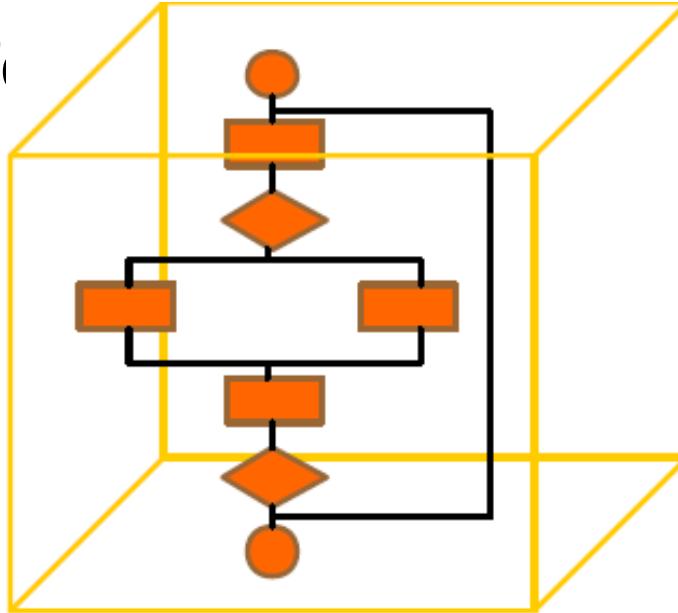


- Functional Testing (Black Box Testing):
  - Test dựa trên mô tả, chúng ta xem xét phần mềm với các dữ liệu đầu vào và đầu ra mà không cần biết cấu trúc của phần mềm ra sao. Nghĩa là tester sẽ tập trung vào **những gì mà phần mềm làm**, không cần biết phần mềm làm như thế nào.
  - Ưu điểm:
    - Không phụ thuộc vào việc thực hiện phần mềm
    - Việc phát triển test case có thể diễn ra song song với quá trình thực hiện phần mềm → Rút ngắn thời gian thực hiện dự án

# Các phương pháp kiểm thử (2)



- Structural Testing (White Box Testing):
  - Test dựa trên cấu trúc còn được gọi là white-box hay glass-box bởi vì nó đòi hỏi sự hiểu biết về cấu trúc của phần mềm, nghĩa là phần mềm hoạt động như thế nào



# Các phương pháp kiểm thử (3)



- Experience Testing (Test dựa trên kinh nghiệm)
  - Kỹ thuật này đòi hỏi sự hiểu biết, kỹ năng và kinh nghiệm của người test.
  - Dựa vào những kinh nghiệm thu thập được từ những hệ thống trước đó, tester có thể dễ dàng nhìn thấy được những điểm sai trong chương trình.

# Tổng quan về kiểm thử hộp đen



- Phương pháp kiểm thử hộp đen: coi hệ thống là một hộp đen, không thể thấy được cấu trúc logic bên trong. Người làm kiểm thử tập trung vào các yêu cầu chức năng của phần mềm dựa trên các dữ liệu lấy từ đặc tả
- Đặc trưng:
  - Nhầm thuyết minh: các chức năng phần mềm đủ & vận hành đúng
  - Thực hiện các phép thử qua giao diện



# Tổng quan về kiểm thử hộp đen



- Kiểm thử hộp đen nhằm tìm ra các loại sai:
  - Chức năng thiếu hoặc không đúng đắn.
  - Sai về giao diện.
  - Sai trong cấu trúc hoặc trong truy cập dữ liệu ngoài.
  - Sai thực thi chức năng.
  - Sai khởi đầu hoặc kết thúc mô đun.

# Các kỹ thuật kiểm thử hộp đen



- Kỹ thuật phân lớp tương đương (Equivalence Class Testing)
- Kỹ thuật dựa trên giá trị biên (Boundary Value Testing)
- Kỹ thuật dựa trên bảng quyết định (Decision Table-Based Testing)
- Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (causes-effects)
- ...

# Các kỹ thuật kiểm thử hộp đen (1)

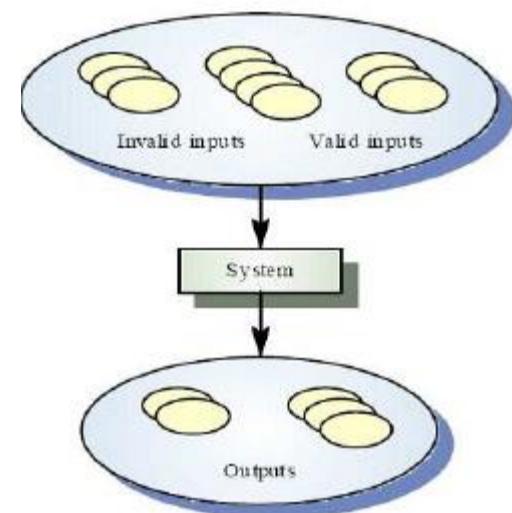


- Kỹ thuật phân lớp tương đương (Equivalence Class Testing)
- Kỹ thuật dựa trên giá trị biên (Boundary Value Testing)
- Kỹ thuật dựa trên bảng quyết định (Decision Table-Based Testing)
- Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (causes-effects)
- ...

# Kỹ thuật phân lớp tương đương



- Ý tưởng: Chia miền vào chương trình thành các lớp dữ liệu. Xác định đầu vào hợp lệ và không hợp lệ để lập các ca kiểm thử theo các lớp đó
- Mỗi lớp dùng để kiểm thử một chức năng, gọi là lớp tương đương.
- Thay vì kiểm tra tất cả các giá trị đầu vào, có thể lựa chọn từ đầu vào cho riêng từng lớp



# Kỹ thuật phân lớp tương đương



- Nguyên tắc xác định lớp tương đương:
  - Nếu điều kiện đầu vào định rõ giới hạn của một mảng, hoặc một giá trị xác định thì chia vùng tương đương thành:
    - Một lớp tương đương hợp lệ
    - Hai lớp không hợp lệ
    - Một lớp đặc biệt (nếu có)
  - Nếu điều kiện đầu vào chỉ định là một tập giá trị, hoặc xác định là một kiểu đúng sai thì chia vùng tương đương thành :
    - Một lớp tương đương hợp lệ.
    - Một lớp tương đương không hợp lệ.
    - Một lớp đặc biệt (nếu có)

# Kỹ thuật phân lớp tương đương



- Ví dụ

STT	Trường hợp của điều kiện đầu vào	Lớp tương đương	
		Hợp lệ	Không hợp lệ
1	Là 1 khoảng các giá trị Ví dụ: [3,10]	1 $3 < x < 10$	2 $x < 3$ hoặc $x > 10$
	Là 1 giá trị cụ thể Ví dụ: 6	1 $x = 6$	2 $x < 6$ hoặc $x > 6$
3	Là 1 tập hợp các giá trị Ví dụ: $A = \{1, 5, 4, 8, 6\}$	1 $x$	1
	Là 1 giá trị Boolean	1	1

# Kỹ thuật phân lớp tương đương



- Ví dụ: Một textbox chỉ cho phép nhập số nguyên từ 1 đến 100
  - Ta không thể nhập tất cả các giá trị từ 1 đến 100
- **Ý tưởng của kỹ thuật này:** Chia (partition) đầu vào thành những nhóm nhóm tương đương nhau (equivalence).
- **Giảm đáng kể số lượng test case** cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện

# Kỹ thuật phân lớp tương đương



- Có hai yếu tố ảnh hưởng đến việc thiết kế test case
  - Dựa trên giả định (Assumption)
    - Single fault assumption → Weak ECT (Equivalence Class Testing)
    - Multiple fault assumption → Strong ECT
  - Dựa trên loại dữ liệu inputs
    - Kiểm thử trên dữ liệu hợp lệ → Normal ECT
    - Kiểm thử trên dữ liệu không hợp lệ → Robust ECT

Assumption Data	Single Fault	Multiple Faults
Valid	Weak Normal	Strong Normal
Invalid	Weak Robust	Strong Robust

# Kỹ thuật phân lớp tương đương

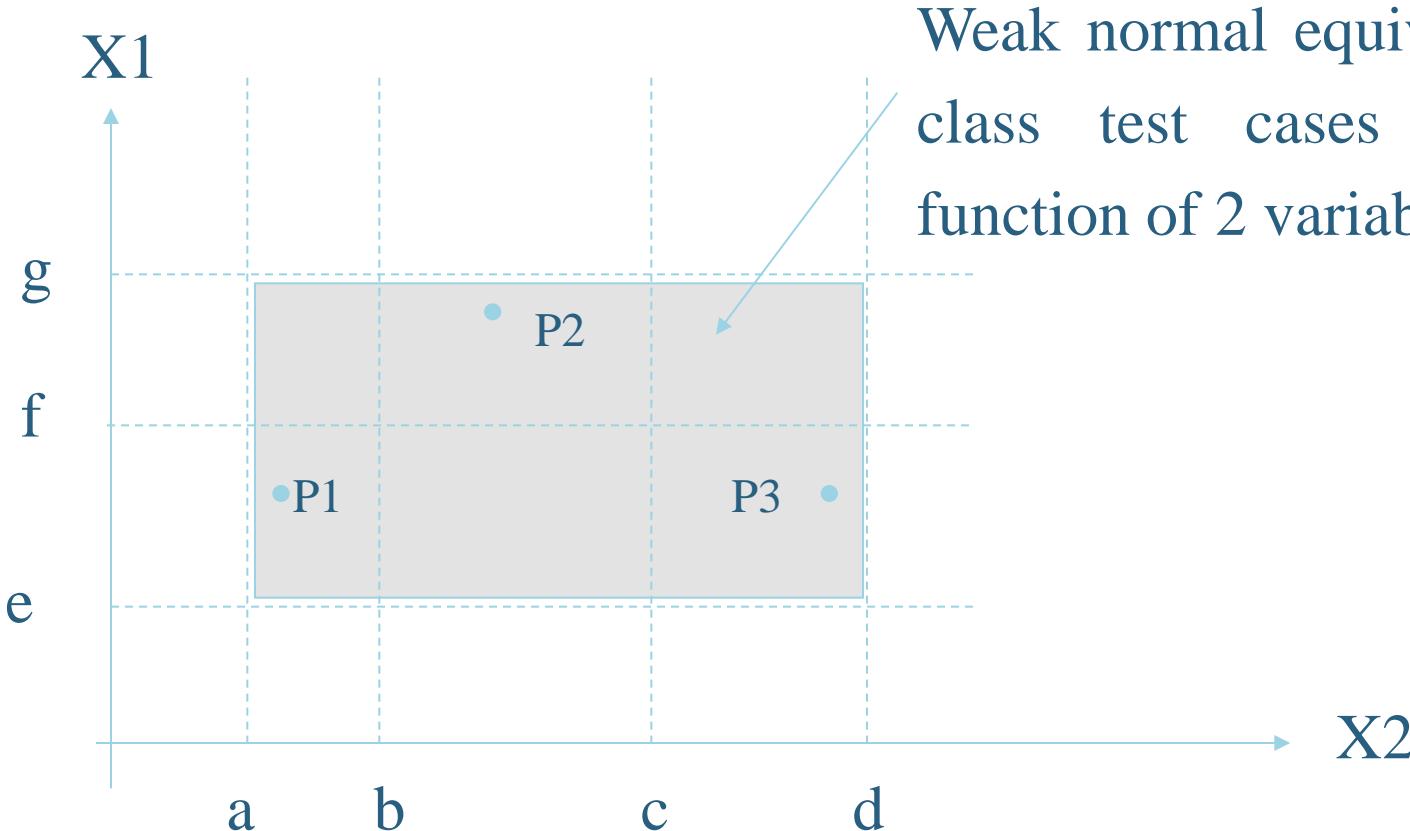


- Weak Normal Equivalence Class Testing
- Strong Normal Equivalence Class Testing
- Weak Robust Equivalence Class Testing
- Strong Robust Equivalence Class Testing



- Dựa trên Single Fault Assumption
  - Một failure ít khi nào là kết quả của 2 hay nhiều faults xảy ra cùng 1 lúc
- Ví dụ:
  - $e \leq x_1 \leq g$ ,  $x_1$  có 2 lớp tương đương  $[e, f)$   $[f, g]$
  - $a \leq x_2 \leq d$ ,  $x_2$  có 3 lớp tương đương  $[a, b)$   $[b, c)$ ,  $[c, d]$

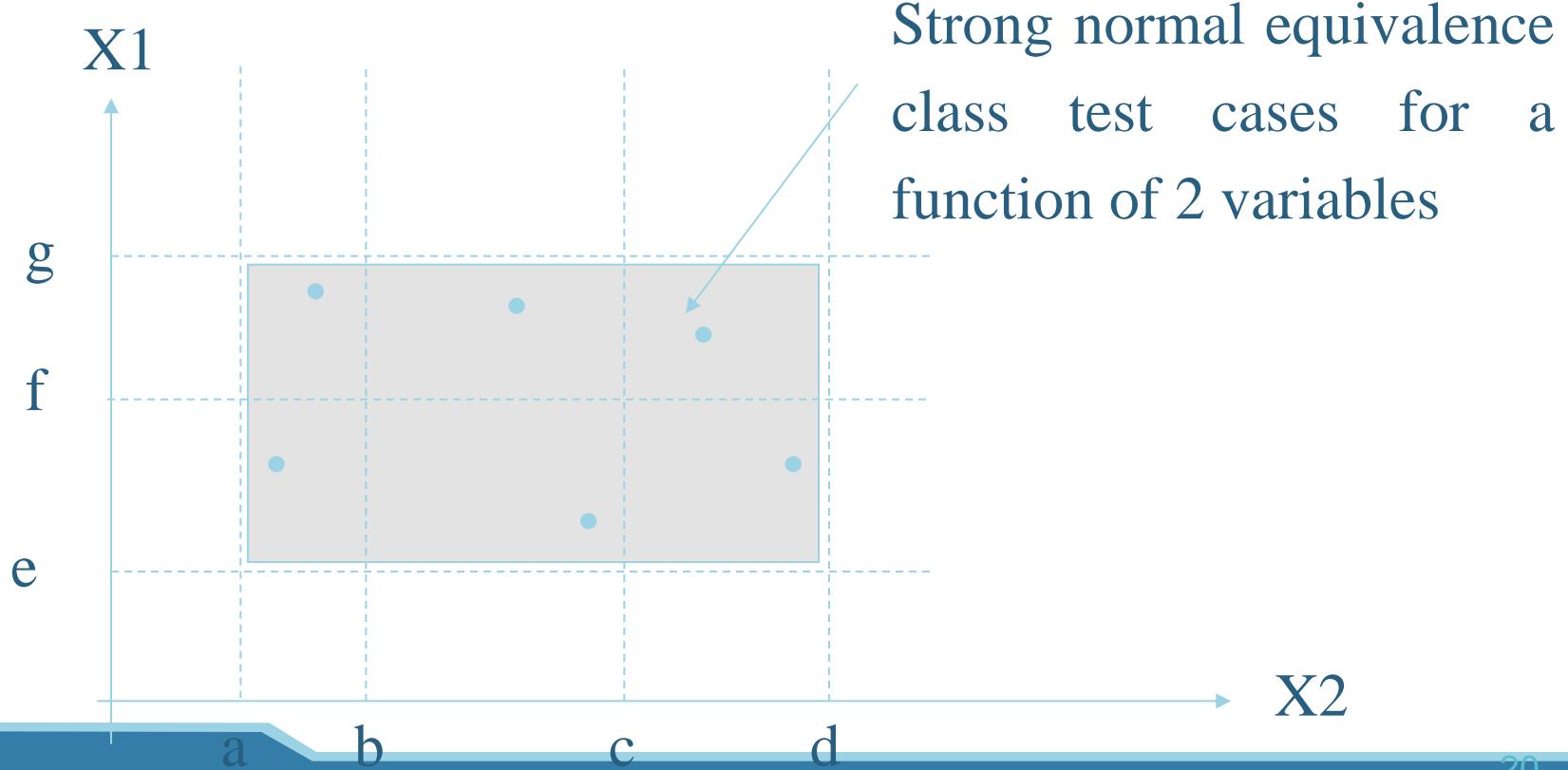
# Weak Normal Equivalence Class Testing



# Strong Normal Equivalence Class Testing



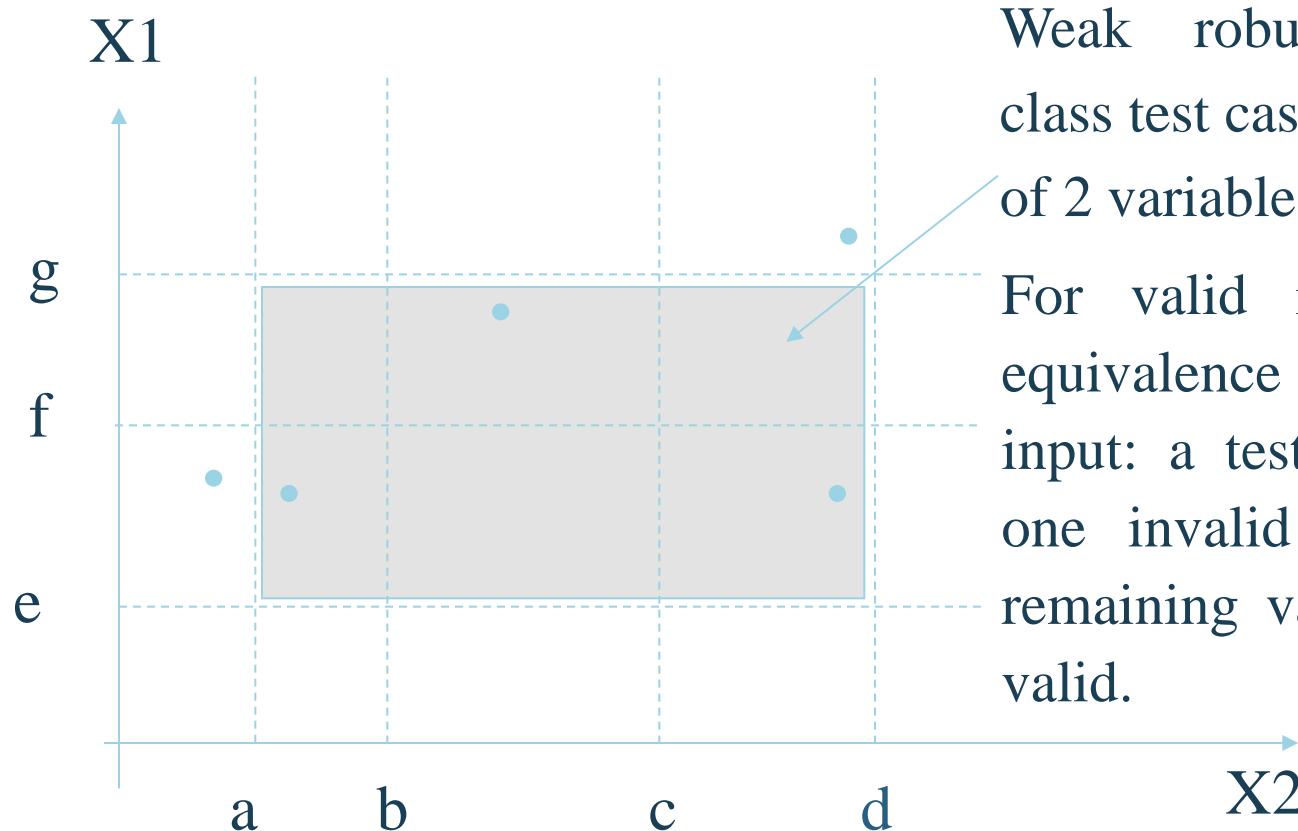
- Dựa trên Multiple Fault Assumption
  - Một failure có thể là kết quả của 2 hay nhiều faults xảy ra cùng 1 lúc



# Weak Robust Equivalence Class Testing



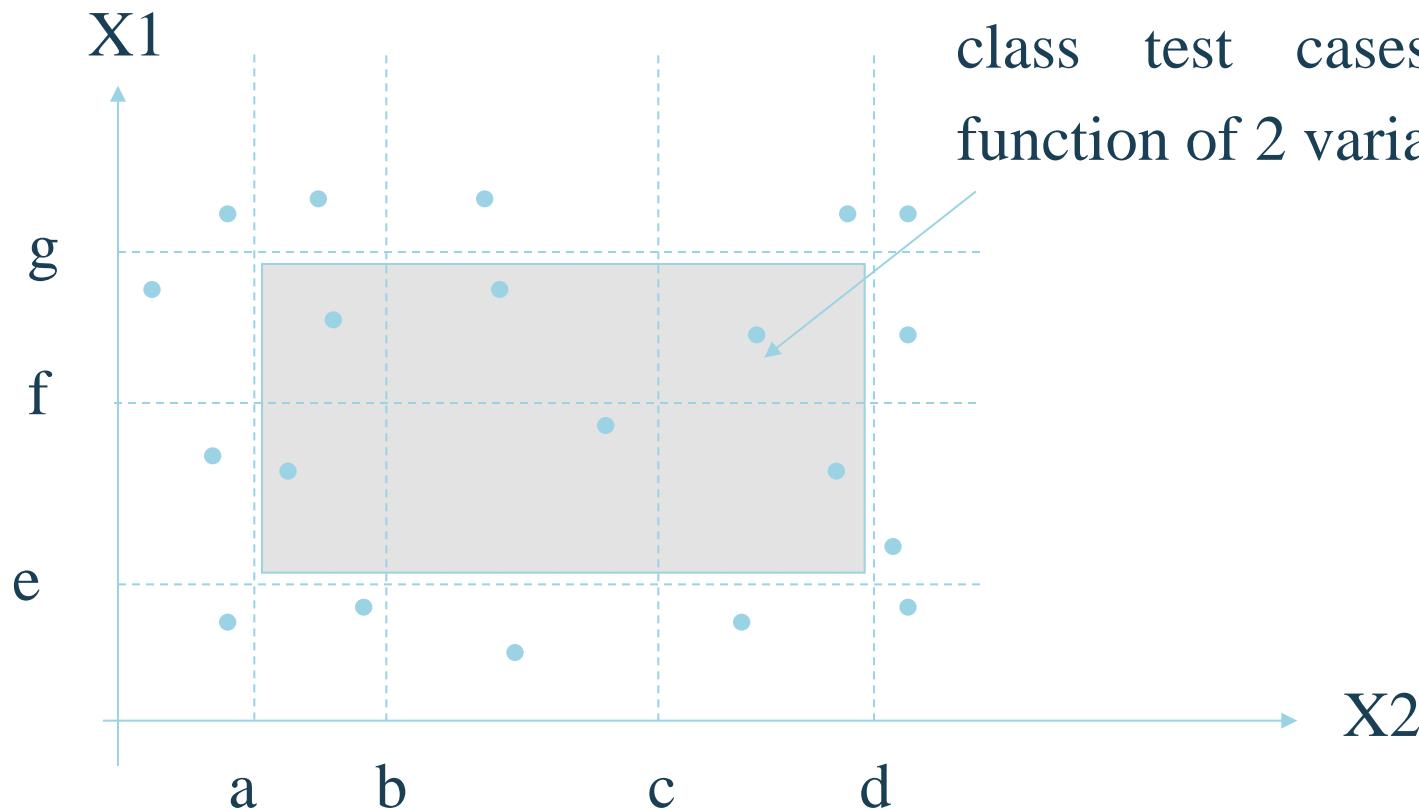
- Tương tự Weak Equivalence Class Testing, tuy nhiên test thêm trường hợp 1 biến với giá trị không hợp lệ



# Strong Robust Equivalence Class Testing



Strong robust equivalence class test cases for a function of 2 variables



# Ví dụ 1



- TPPM “Quản lý hồ sơ nhân lực” với đặc tả chức năng như sau: sau mỗi lần nhận 1 hồ sơ xin việc, TPPM sẽ ra quyết định ban đầu dựa vào tuổi của ứng viên theo bảng sau:
    - **Tuổi ứng viên Kết quả sơ bộ**
      - 0-15 Không thuê
      - 16-17 Thuê dạng bán thời gian
      - 18-54 Thuê toàn thời gian
      - 55-99 Không thuê
  - Bằng phương pháp phân hoạch tương đương và phân tích giá trị biên, hãy thiết kế các trường hợp kiểm thử cho TPPM trên.

# Ví dụ 1



- Bảng phân lớp tương đương

Tuổi	<0	0-15	16-17	18-54	55-99	>99
Kết quả	Nhập dữ liệu sai	Ko Thuê	Thuê bán thời gian	Thuê toàn thời gian	Ko Thuê	Nhập dữ liệu sai
Lớp tương đương	Ko Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Ko Hợp lệ
Đánh dấu	I1	V1	V2	V3	V4	I2

- Từ mỗi lớp tương đương, xét các biên cần kiểm thử
  - Lớp I1: {-2,-1,0} Lớp I2: {99,100,101}
  - Lớp V1: {-1,0,1} {14,15,16}
  - Lớp V2: { 15,16,17} {16,17,18}
  - Lớp V3: {17,18,19}, {53,54,55}
  - Lớp V4: {54,55,56}, {98,99,100}

## Ví dụ 1:



- Xét trong các trường hợp trên thấy có nhiều giá trị test case trùng nhau, nếu loại bỏ các test case trùng nhau đó thì ta còn: -1, 0, 1, 14, 15, 16, 17, 18, 19, 53, 54, 55, 56, 98, 99, 100.
- Do trường hợp -1 và 100 đã nằm tại biên của hai lớp tương đương ko hợp lệ nên ta ko xét trường hợp -2, và 101 ở lớp này.
- 16 ca kiểm thử được thiết kế như sau:

# Ví dụ 1: các ca kiểm thử



TC	Đầu vào	Đầu ra mong đợi
1	-1	Nhập sai dữ liệu
2	0	Không thuê
3	1	Không thuê
4	14	Không thuê
5	15	Không thuê
6	16	Thuê bán thời gian
7	17	Thuê bán thời gian
8	18	Thuê toàn thời gian

TC	Đầu vào	Đầu ra mong đợi
9	19	Thuê toàn thời gian
10	53	Thuê toàn thời gian
11	54	Thuê toàn thời gian
12	55	Không thuê
13	56	Không thuê
14	98	Không thuê
15	99	Không thuê
16	100	Nhập sai dữ liệu

# Ví dụ 2: Ứng dụng vay nợ



Customer Name	<input type="text"/>	2-64 chars.
Account number	<input type="text"/>	6 digits, 1st non-zero
Loan amount requested	<input type="text"/>	£500 to £9000
Term of loan	<input type="text"/>	1 to 30 years
Monthly repayment	<input type="text"/>	Minimum £10
Term:		
Repayment:		
Interest rate:		
Total paid back:		

# Các kỹ thuật kiểm thử hộp đen (2)



- Kỹ thuật phân lớp tương đương (Equivalence Class Testing)
- Kỹ thuật dựa trên giá trị biên (Boundary Value Testing)
- Kỹ thuật dựa trên bảng quyết định (Decision Table-Based Testing)
- Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (causes-effects)
- ...

# Kỹ thuật phân tích giá trị biên

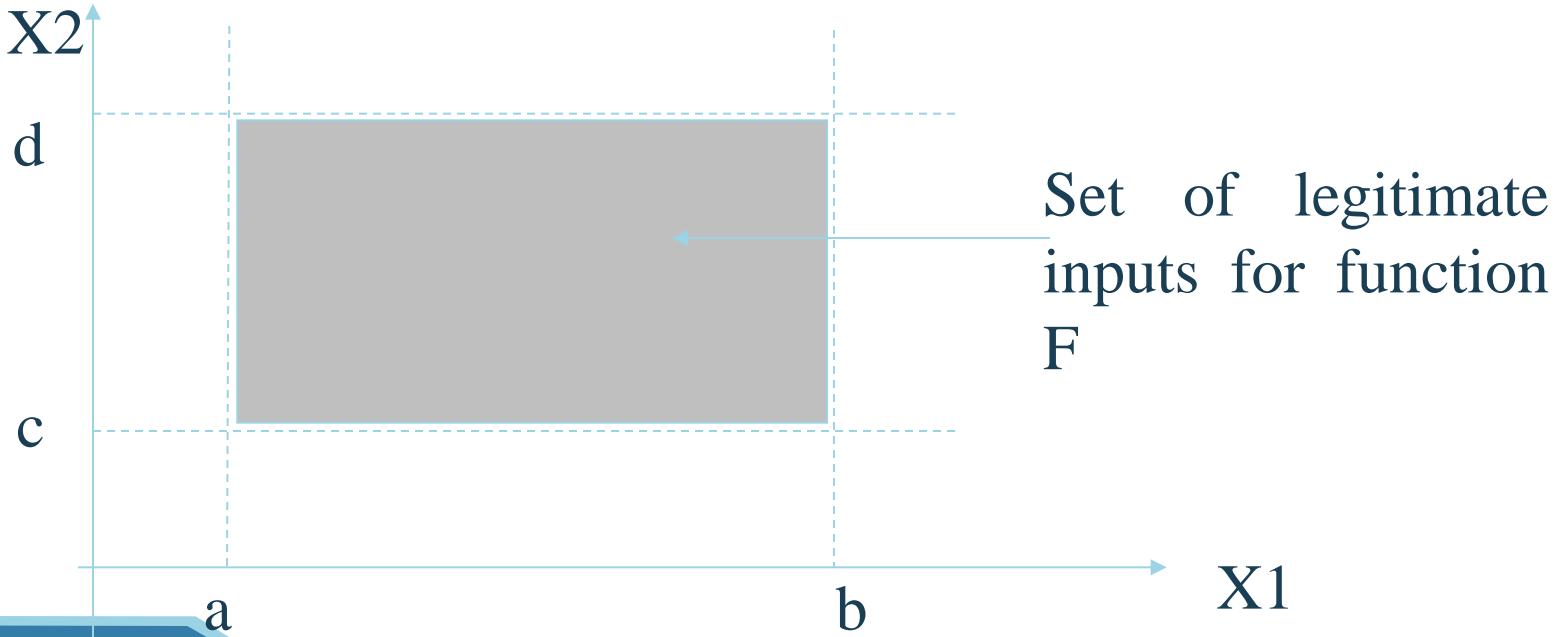


- Phân tích giá trị biên - Boundary Value Analysis
- Thường được áp dụng đối với các đối số của một phương thức
- Tập trung vào việc kiểm thử các giá trị biên của miền giá trị inputs để thiết kế test case do “lỗi thường tiềm ẩn lại các ngõ ngách và tập hợp tại biên” ( Beizer )
- BVA hiệu quả nhất trong trường hợp “các đối số đầu vào (input variables) độc lập với nhau và mỗi đối số đều có một miền giá trị hữu hạn”

# Kỹ thuật phân tích giá trị biên



- Giả sử hàm  $F$  có hai biến  $X_1, X_2$  như sau:
  - $a \leq X_1 \leq b$
  - $c \leq X_2 \leq d$
- Input domain of a function of two variables:



# Một số kỹ thuật kiểm thử giá trị biên



- Standard BVA ( Boundary Value Analysis )
- Robustness testing
- Worst-case testing
- Robust worst-case testing

# Standard BVA

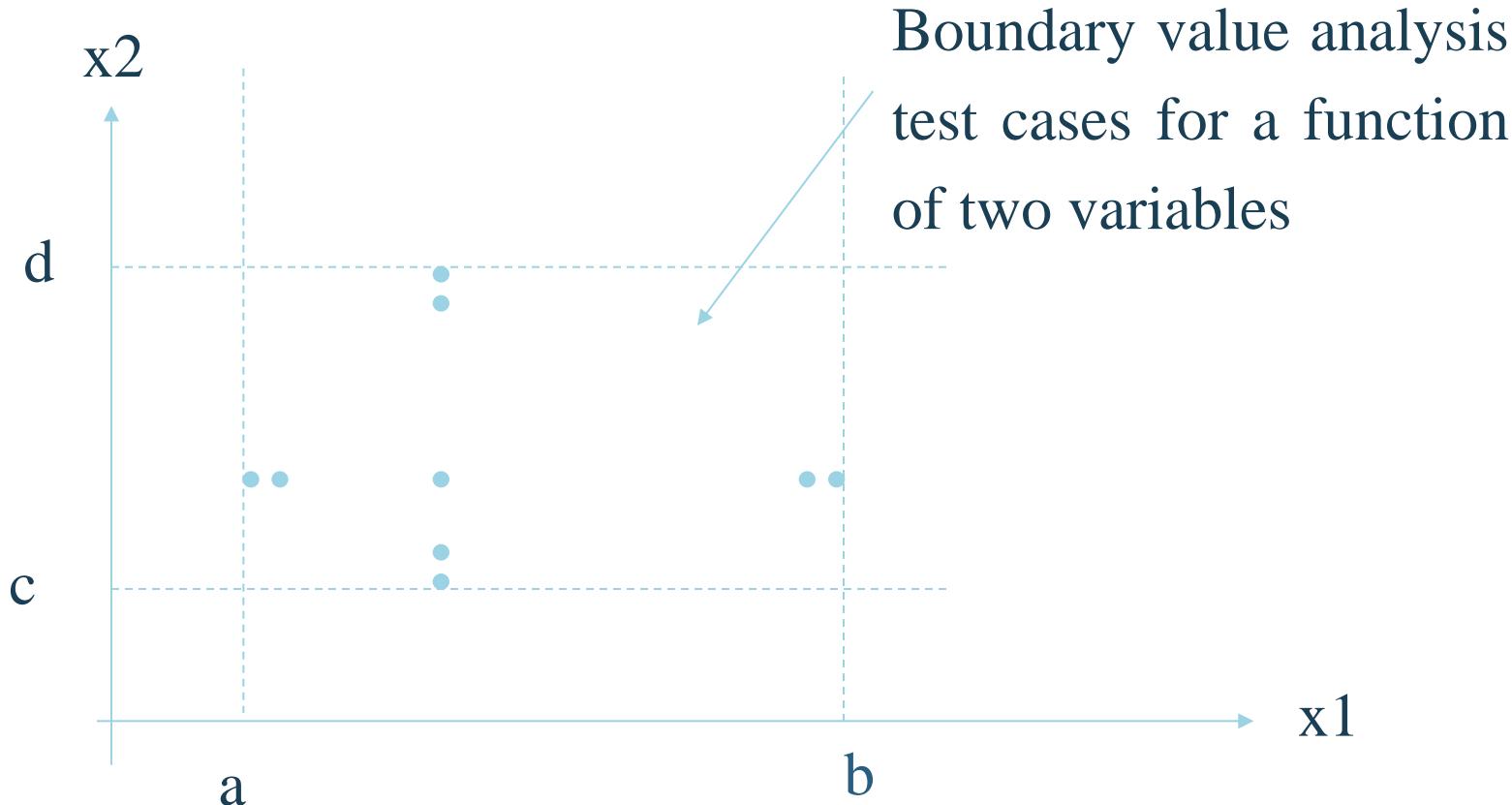


- Giả sử biến x có **miền giá trị [min,max]**
  - Các giá trị được chọn để kiểm tra
    - Min - Minimal
    - Min+ - Just above Minimal
    - Nom - Average
    - Max- - Just below Maximum
    - Max - Maximum

# Kỹ thuật phân tích trên giá trị biên



- Số test case là  $4n+1$ , với  $n$  là số lượng biến



# Robustness Testing

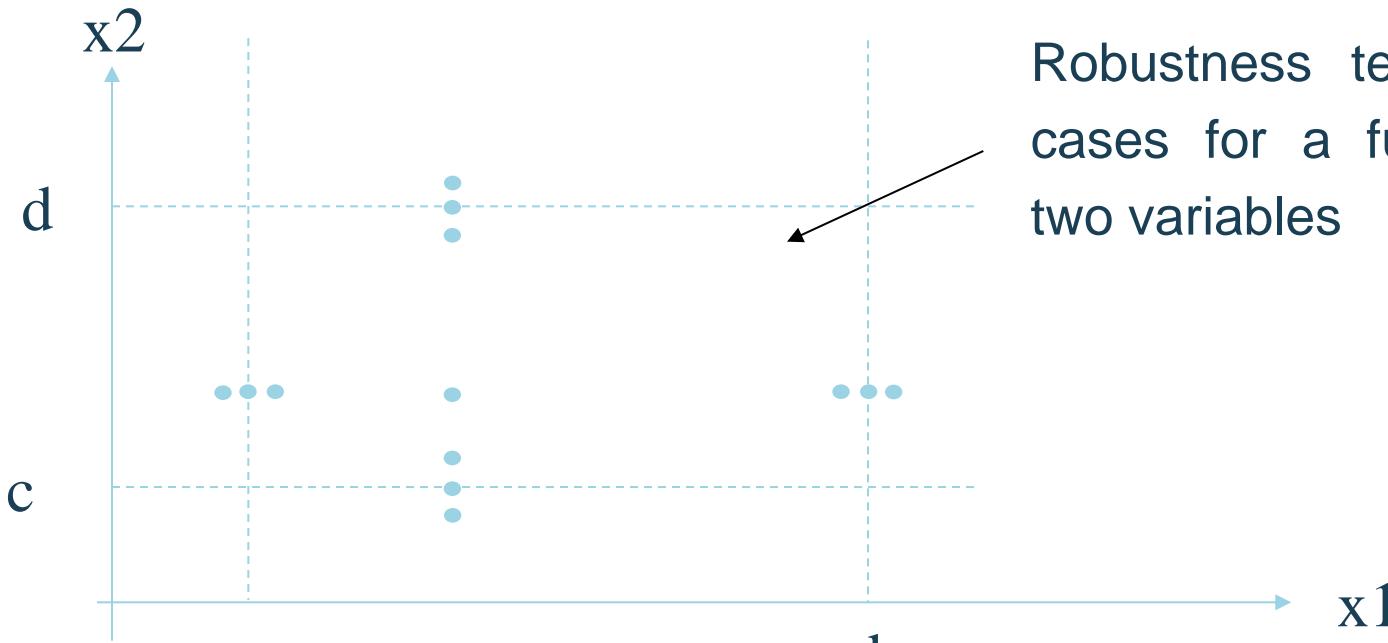


- Mở rộng của Standard BVA
- Kiểm thử cả hai trường hợp:
  - Input variable hợp lệ (clean test cases)
    - Kiểm thử tương tự như Standard BVA trên các giá trị (min, min+, average, max-, max)
  - Input variable không hợp lệ (dirty test cases)
    - Kiểm thử trên 2 giá trị: min-, max+ (nằm ngoài miền giá trị hợp lệ)

# Robustness Testing



- Số lượng test case là  $6n + 1$ , với  $n$  là số lượng biến



- Tập trung <sup>a</sup> vào việc kiểm thử trên các giá trị không hợp lệ và đòi hỏi ứng dụng phải xử lý ngoại lệ một cách đầy đủ

# Worst-case testing

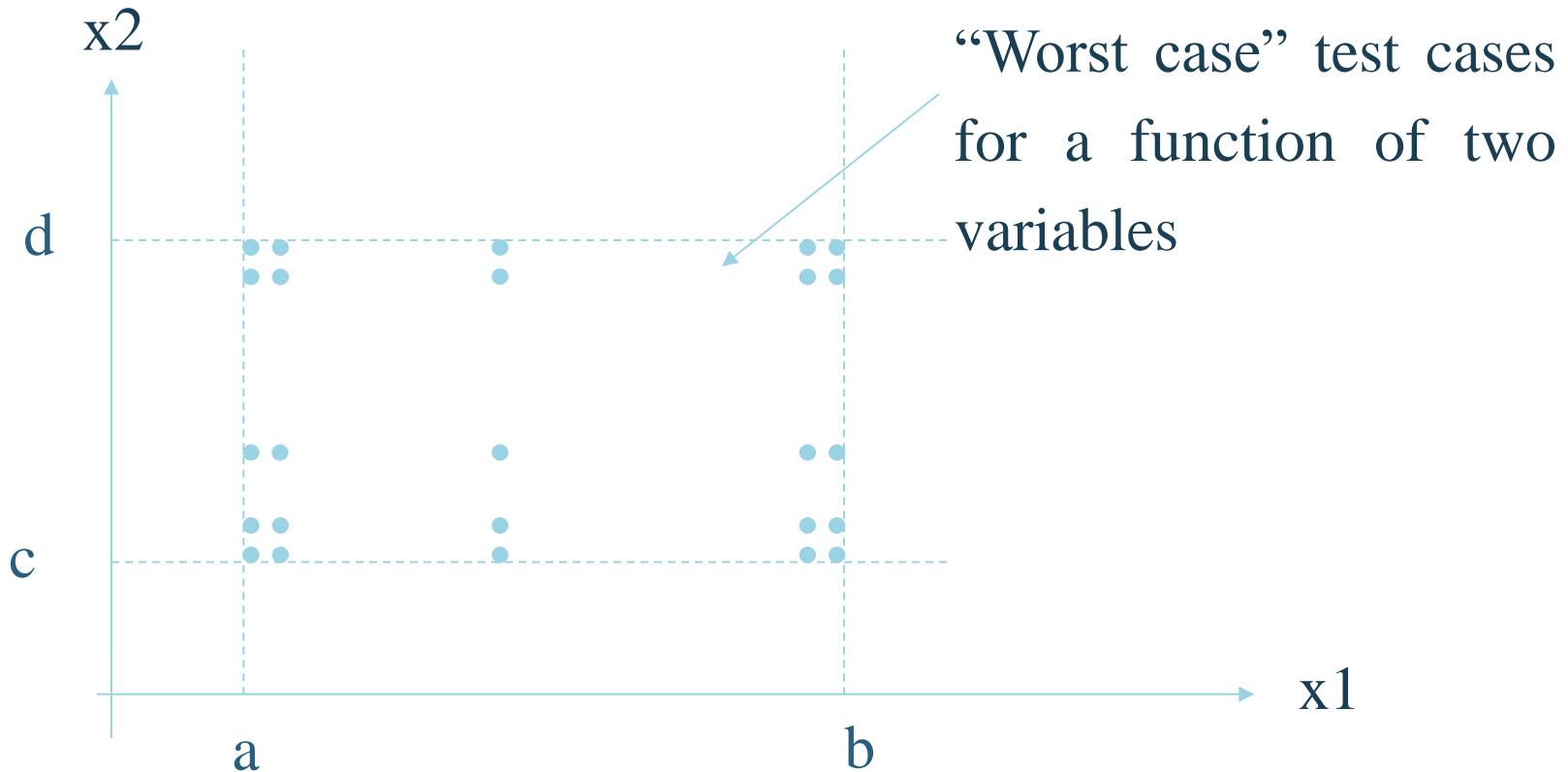


- Dựa trên **Multiple Fault Assumption** để thiết kế test case
- Các biến sẽ được kiểm tra đồng thời tại biên để dò lỗi
- Chúng ta không kiểm thử tại các giá trị không hợp lệ

# Worst-case testing



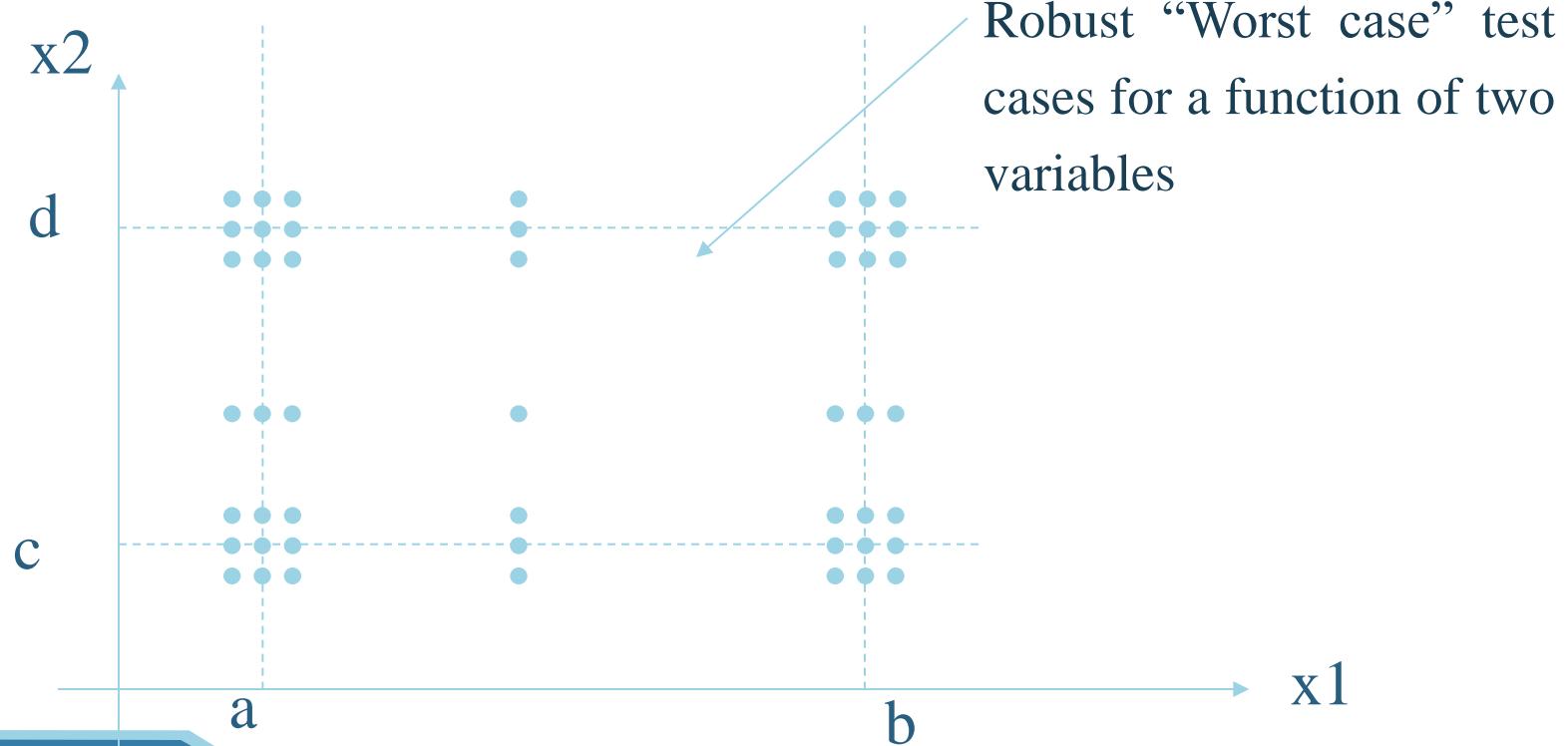
- Số lượng test case là  $5^n$ , với n là số biến



# Robust worst-case testing



- Tương tự Worst-case Testing nhưng kiểm tra thêm tại các giá trị không hợp lệ của input variables (min-, max+)
- Số lượng test case là  $7^n$ , với n là số biến



# Ví dụ hàm kiểm tra tam giác



- Ràng buộc:  $1 \leq a, b, c \leq 200$ .
- Áp dụng Standard BVA (số test case  $4*3 + 1 = 13$ )
  - $\text{min} = 1$
  - $\text{min+} = 2$
  - $\text{nom} = 100$
  - $\text{max-} = 199$
  - $\text{max} = 200$

Boundary Value Analysis Test Cases				
Case	a	b	c	Expected Output
1	100	100	1	Isosceles
2	100	100	2	Isosceles
3	100	100	100	Equilateral
4	100	100	199	Isosceles
5	100	100	200	Not a Triangle
6	100	1	100	Isosceles
7	100	2	100	Isosceles
8	100	199	100	Isosceles
9	100	200	100	Not a Triangle
10	1	100	100	Isosceles
11	2	100	100	Isosceles
12	199	100	100	Isosceles
13	200	100	100	Not a Triangle

# Ví dụ hàm kiểm tra tam giác



- Áp dụng Worst-case testing

Worst Case Test Cases (60 of 125)				
Case	a	b	c	Expected Output
1	1	1	1	Equilateral
2	1	1	2	Not a Triangle
3	1	1	100	Not a Triangle
4	1	1	199	Not a Triangle
5	1	1	200	Not a Triangle
6	1	2	1	Not a Triangle
7	1	2	2	Isosceles
8	1	2	100	Not a Triangle
9	1	2	199	Not a Triangle
10	1	2	200	Not a Triangle
11	1	100	1	Not a Triangle
12	1	100	2	Not a Triangle
13	1	100	100	Isosceles
14	1	100	199	Not a Triangle
15	1	100	200	Not a Triangle
16	1	199	1	Not a Triangle
17	1	199	2	Not a Triangle
18	1	199	100	Not a Triangle
19	1	199	199	Isosceles
20	1	199	200	Not a Triangle
21	1	200	1	Not a Triangle
22	1	200	2	Not a Triangle
23	1	200	100	Not a Triangle
24	1	200	199	Not a Triangle
25	1	200	200	Isosceles
26	2	1	1	Not a Triangle
27	2	1	2	Isosceles
28	2	1	100	Not a Triangle
29	2	1	199	Not a Triangle
30	2	1	200	Not a Triangle
31	2	2	1	Isosceles
32	2	2	2	Equilateral
33	2	2	100	Not a Triangle
34	2	2	199	Not a Triangle
35	2	2	200	Not a Triangle
36	2	100	1	Not a Triangle
37	2	100	2	Not a Triangle
38	2	100	100	Isosceles
39	2	100	199	Not a Triangle
40	2	100	200	Not a Triangle
41	2	199	1	Not a Triangle
42	2	199	2	Not a Triangle
43	2	199	100	Not a Triangle
44	2	199	199	Isosceles
45	2	199	200	Scalene
46	2	200	1	Not a Triangle
47	2	200	2	Not a Triangle
48	2	200	100	Not a Triangle
49	2	200	199	Scalene
50	2	200	200	Isosceles
51	100	1	1	Not a Triangle
52	100	1	2	Not a Triangle
53	100	1	100	Isosceles
54	100	1	199	Not a Triangle
55	100	1	200	Not a Triangle
56	100	2	1	Not a Triangle
57	100	2	2	Not a Triangle
58	100	2	100	Isosceles
59	100	2	199	Not a Triangle
60	100	2	200	Not a Triangle

# Ví dụ hàm tìm ngày kế tiếp



- Bài toán tìm ngày kế tiếp với các ràng buộc:
  - $1 \leq \text{Day} \leq 31$ .
  - $1 \leq \text{month} \leq 12$ .
  - $1812 \leq \text{Year} \leq 2012$
- Áp dụng Standard BVA (số test case  $4^*3 + 1 = 13$ )

# Ví dụ hàm tìm ngày kế tiếp



<u>month</u>	<u>day</u>
min = 1	min = 1
min+ = 2	min+ = 2
nom = 6	nom = 15
max- = 11	max- = 30
max = 12	max = 31

<u>year</u>
min = 1812
min+ = 1813
nom = 1912
max- = 2011
max = 2012

Boundary Value Analysis Test Cases

Case	month	day	year	Expected Output
1	6	15	1812	June 16, 1812
2	6	15	1813	June 16, 1813
3	6	15	1912	June 16, 1912
4	6	15	2011	June 16, 2011
5	6	15	2012	June 16, 2012
6	6	1	1912	June 2, 1912
7	6	2	1912	June 3, 1912
8	6	30	1912	July 1, 1912
9	6	31	1912	error
10	1	15	1912	January 16, 1912
11	2	15	1912	February 16, 1912
12	11	15	1912	November 16, 1912
13	12	15	1912	December 16, 1912

# Ví dụ hàm tìm ngày kế tiếp



- Áp dụng Worst-case testing, Số lượng test case:  $5^3$

Worst Case Test Cases (60 of 125)				
Case	month	day	year	Expected Output
1	1	1	1812	January 2, 1812
2	1	1	1813	January 2, 1813
3	1	1	1912	January 2, 1912
4	1	1	2011	January 2, 2011
5	1	1	2012	January 2, 2012
6	1	2	1812	January 3, 1812
7	1	2	1813	January 3, 1813
8	1	2	1912	January 3, 1912
9	1	2	2011	January 3, 2011
10	1	2	2012	January 3, 2012
11	1	15	1812	January 16, 1812
12	1	15	1813	January 16, 1813
13	1	15	1912	January 16, 1912
14	1	15	2011	January 16, 2011
15	1	15	2012	January 16, 2012
16	1	30	1812	January 31, 1812
17	1	30	1813	January 31, 1813
18	1	30	1912	January 31, 1912
19	1	30	2011	January 31, 2011
20	1	30	2012	January 31, 2012
21	1	31	1812	February 1, 1812
22	1	31	1813	February 1, 1813
23	1	31	1912	February 1, 1912
24	1	31	2011	February 1, 2011
25	1	31	2012	February 1, 2012
26	2	1	1812	February 2, 1812
27	2	1	1813	February 2, 1813
28	2	1	1912	February 2, 1912
29	2	1	2011	February 2, 2011
30	2	1	2012	February 2, 2012
31	2	2	1812	February 3, 1812
32	2	2	1813	February 3, 1813
33	2	2	1912	February 3, 1912
34	2	2	2011	February 3, 2011
35	2	2	2012	February 3, 2012
36	2	15	1812	February 16, 1812
37	2	15	1813	February 16, 1813
38	2	15	1912	February 16, 1912
39	2	15	2011	February 16, 2011
40	2	15	2012	February 16, 2012
41	2	30	1812	error
42	2	30	1813	error
43	2	30	1912	error
44	2	30	2011	error
45	2	30	2012	error
46	2	31	1812	error
47	2	31	1813	error
48	2	31	1912	error
49	2	31	2011	error
50	2	31	2012	error
51	6	1	1812	June 2, 1812
52	6	1	1813	June 2, 1813
53	6	1	1912	June 2, 1912
54	6	1	2011	June 2, 2011
55	6	1	2012	June 2, 2012
56	6	2	1812	June 3, 1812
57	6	2	1813	June 3, 1813
58	6	2	1912	June 3, 1912
59	6	2	2011	June 3, 2011
60	6	2	2012	June 3, 2012

# Các kỹ thuật kiểm thử hộp đen (3)



- Kỹ thuật dựa trên giá trị biên (Boundary Value Testing)
- Kỹ thuật phân lớp tương đương (Equivalence Class Testing)
- Kỹ thuật dựa trên bảng quyết định (Decision Table-Based Testing)
- Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (cause-effect graphing)
- ...

# Bảng quyết định



- Là kỹ thuật được áp dụng trong nhiều lĩnh vực:
  - Phân tích logic trong các hoạt động nghiệp vụ
  - Lập trình
  - Kiểm thử
  - ...
- Làm **giảm số lượng test case không cần thiết** so với 2 kỹ thuật Equivalence Class và Boundary Value Analysis vì nó **loại trừ các phép kết hợp không cần thiết** giữa các input variables

# Bảng quyết định



- ❖ Liệt kê các **nguyên nhân (cause)** – **kết quả (effect)** trong 1 ma trận. Mỗi cột trong ma trận đại diện cho 1 phép kết hợp giữa các cause trong việc tạo ra 1 effect

		Combinations							
Causes	Values	1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X		X	

Cause = Condition

Effect = Actions = Expected Results

# Các bước để tạo ra Bảng quyết định



- Liệt kê tất cả các nguyên nhân (causes) trong bảng quyết định
- Tính tổng số lượng kết hợp giữa các cause
- Điền vào các cột với tất cả các kết hợp có thể có
- Rút bớt số lượng các phép kết hợp dư thừa
- Kiểm tra các phép kết hợp có bao phủ hết mọi trường hợp hay không
- Bổ sung kết quả (effects) vào bảng quyết định

# B1: Liệt kê tất cả các nguyên nhân



Causes	Values	Combinations							
		1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X		X	

- c1:  $a < b+c?$
- c2:  $b < a+c?$
- c3:  $c < a+b?$
- c4:  $a = b?$
- c5:  $a = c?$
- c6:  $b = c?$

- Điene vào các giá trị trong từng causes
  - Gom nhóm các causes có liên quan với nhau
  - Sắp xếp các cause theo thứ tự giảm dần theo độ ưu tiên
- Ví dụ: xét bài toán kiểm tra loại của 1 tam giác dựa vào chiều dài 3 cạnh a, b, c.

## B2: Tính tổng số kết hợp giữa các causes



- Tổng số phép kết hợp  
= (số lượng values của cause 1) \* ... \* (số lượng values của cause n)

c1:  $a < b + c$ ?

c2:  $b < a + c$ ?

c3:  $c < a + b$ ?

c4:  $a = b$ ?

c5:  $a = c$ ?

c6:  $b = c$ ?

Mỗi cause có 2 giá trị true, false

→ Tổng số phép kết hợp =  $2^6 = 64$

# B3: Điene giá trị các cột trong bảng



Causes	Values	Combinations							
		1	2	3	4	5	6	7	8
Cause 1	Y, N	Y	Y	Y	Y	N	N	N	N
Cause 2	Y, N	Y	Y	N	N	Y	Y	N	N
Cause 3	Y, N	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X		X				X	
Effect 2			X			X			X

## • Thuật toán:

- Xác định số lần lặp lại (RF) trong từng giá trị của cause bằng cách lấy tổng số phép kết hợp **còn lại** chia cho số values mà cause có thể nhận
- Điene dữ liệu cho dòng thứ i: điene RF lần **giá trị đầu tiên** của cause i, tiếp theo RF lần **giá trị tiếp theo** của cause i... cho đến khi dòng đầy
- Chuyển sang dòng kế tiếp, quay lại bước 1 và tiếp tục thực hiện

# B3: Điền giá trị các cột trong bảng



- **Ví dụ:**

c1: $a < b+c$ ?	$\xleftarrow{\hspace{3cm}} 32 \xrightarrow{\hspace{3cm}}$	$\xleftarrow{\hspace{3cm}} 32 \xrightarrow{\hspace{3cm}}$
	$F \xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} T$	$T \xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} F$
c2: $b < a+c$ ?	$\xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} F$	$\xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} T$
	$F \xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} T$	$T \xleftarrow{\hspace{1.5cm}} 16 \xrightarrow{\hspace{1.5cm}} F$
c3: $c < a+b$ ?	$\xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} F$	$\xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} T$
	$T \xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} F$	$F \xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} T$
	$F \xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} T$	$T \xleftarrow{\hspace{0.75cm}} 8 \xrightarrow{\hspace{0.75cm}} F$
c4: $a=b$ ?	....	

$$RF = 64 / 2 = 32$$

$$RF = 32 / 2 = 16$$

$$RF = 16 / 2 = 8$$

## B4: Giảm số phép kết hợp



- Duyệt qua tất cả các ô trong từng cột, ô nào mà kết quả của nó không ảnh hưởng đến effect thì đặt giá trị trên ô này là “-” (don't care entry)
- Ghép các cột với nội dung giống nhau thành 1 cột

F	T	T	T	T	T	T	T	T
-	F	T	T	T	T	T	T	T
-	-	F	T	T	T	T	T	T
-	-	-	T	T	T	F	F	F
-	-	-	T	T	F	F	T	F
-	-	-	T	F	T	F	T	F

## B5: Kiểm tra độ bao phủ các phép kết hợp



- Tính rule-count trên từng cột (số lượng phép kết hợp) mà cột này có thể thực hiện
- Với các dòng có giá trị là ‘-’ thì luỹ thừa 2
- Nếu tổng của các rule-count bằng với tổng số kết hợp giữa các cause trong bước 2 thì bảng quyết định là đầy đủ

c1: a < b + c?	F	T	T	T	T	T	T	T	T	T	T
c2: b < a + c?	-	F	T	T	T	T	T	T	T	T	T
c3: c < a + b?	-	-	F	T	T	T	T	T	T	T	T
c4: a = b?	-	-	-	T	T	T	F	F	F	F	F
c5: a = c?	-	-	-	T	T	F	F	T	T	F	F
c6: b = c?	-	-	-	T	F	T	F	T	F	T	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1

## B6: Bổ sung kết quả (effect) vào trong bảng



- Duyệt qua từng cột và check vào kết quả (effect)
- Nhiều cột khác nhau có thể cho ra cùng 1 kết quả giống nhau

c1: $a < b+c?$	F	T	T	T	T	T	T	T	T	T	T
c2: $b < a+c?$	-	F	T	T	T	T	T	T	T	T	T
c3: $c < a+b?$	-	-	F	T	T	T	T	T	T	T	T
c4: $a = b?$	-	-	-	T	T	T	T	F	F	F	F
c5: $a = c?$	-	-	-	T	T	F	F	T	T	F	F
c6: $b = c?$	-	-	-	T	F	T	F	T	F	T	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1
a1: Not a Triangle	X	X	X								
a2: Scalene											X
a3: Isosceles							X		X	X	
a4: Equilateral				X							
a5: Impossible					X	X		X			

# Ví dụ



- Bảng quyết định hoàn chỉnh

c1: $a < b+c?$	F	T	T	T	T	T	T	T	T	T	T	T
c2: $b < a+c?$	-	F	T	T	T	T	T	T	T	T	T	T
c3: $c < a+b?$	-	-	F	T	T	T	T	T	T	T	T	T
c4: $a = b?$	-	-	-	T	T	T	T	F	F	F	F	F
c5: $a = c?$	-	-	-	T	T	F	F	T	T	F	F	F
c6: $b = c?$	-	-	-	T	F	T	F	T	F	T	F	F
Rule Count	32	16	8	1	1	1	1	1	1	1	1	1
a1: Not a Triangle	X	X	X									
a2: Scalene												X
a3: Isosceles								X		X	X	
a4: Equilateral					X							
a5: Impossible						X	X		X			

# Ví dụ 1



- Công ty Honda trao học bổng cho những bạn sinh viên thỏa mãn ít nhất 1 trong 2 điều kiện sau: Là sinh viên giỏi , là cán bộ lớp.
- Nếu thỏa mãn cả 2 điều kiện sẽ được học bổng 600\$, nếu thỏa mãn là sinh viên giỏi được học bổng 400\$, nếu là cán bộ lớp được học bổng là 300\$.
- Lập bảng hỗ trợ quyết định để thiết kế các ca kiểm thử

# VD1: Bảng hỗ trợ quyết định



		Luật 1	Luật 2	Luật 3	Luật 4
Điều kiện	Là cán bộ lớp	Y	Y	N	N
	Là học sinh giỏi	Y	N	Y	N
Hành động	Học bổng	600\$	300\$	400\$	0\$

Các ca kiểm thử

Ca	Đầu vào	Đầu ra mong đợi
1	Là cán bộ lớp, là sv giỏi	600\$
2	Là cán bộ lớp, ko fai là sv giỏi	300\$
3	Ko phải cán bộ lớp, là sv giỏi	400\$
4	Ko phải cán bộ lớp, ko phải sv giỏi	0\$

## VD 2



- Chương trình quản lý tiền vé và số lượng mũ phát cho khách hàng vào thăm quan bảo tàng được mô tả như sau:
- Vé bán có các mức sau: đối với trẻ em dưới 5 tuổi được miễn phí, đối tượng từ 5 tuổi tới 65 tuổi phải trả 20\$/vé, đối tượng lớn hơn 65 tuổi phải trả 10\$/vé.
- Chương trình tặng mũ cho khách: với những đối tượng là nữ sẽ tặng mũ hồng, đối tượng là nam sẽ tặng mũ xanh
- Dùng bảng hỗ trợ quyết định xây dựng các ca kiểm thử cho chương trình trên

# VD2



		Luật 1	Luật 2	Luật 3	Luật 4	Luật 5	Luật 6
Điều kiện	Tuổi <5	Y	Y				
	5<=tuổi<=65			Y	Y		
	Tuổi >65					Y	Y
	Giới tính	Y	N	Y	N	Y	N
Hành động	Miễn phí	Y	Y				
	Giá 10 \$					Y	Y
	Giá 20\$			Y	Y		
	Tặng mũ xanh	Y		Y		Y	
	Tặng mũ hồng		Y		Y		Y

## VD2



- Các ca kiểm thử dựa trên bảng hỗ trợ quyết định

Ca	Đầu vào	Đầu ra mong đợi
1	Tuổi <5, giới tính: Nữ	Miễn phí vé vào, tặng mũ đỏ
2	Tuổi <5, giới tính: Nam	Miễn phí vé vào, tặng mũ xanh
3	5<= tuổi<= 65, giới tính: Nữ	Giá vé 20\$, tặng mũ đỏ
4	5<= tuổi<= 65, giới tính: Nam	Giá vé 20\$, tặng mũ xanh
5	Tuổi >65, giới tính: Nữ	Giá vé 10\$, tặng mũ đỏ
6	Tuổi >65, giới tính: Nam	Giá vé 10\$, tặng mũ xanh

# Các kỹ thuật kiểm thử hộp đen (4)



- Kỹ thuật dựa trên giá trị biên (Boundary Value Testing)
- Kỹ thuật phân lớp tương đương (Equivalence Class Testing)
- Kỹ thuật dựa trên bảng quyết định (Decision Table-Based Testing)
- Kỹ thuật dựa trên đồ thị nguyên nhân – kết quả (causes-effects)
- ...

# Đồ thị nguyên nhân – kết quả



- Là kỹ thuật thiết kế test case dựa trên đồ thị
- Tập trung vào việc xác định các mối kết hợp giữa các conditions và kết quả mà các mối kết hợp này mang lại

# Các bước xây dựng đồ thị

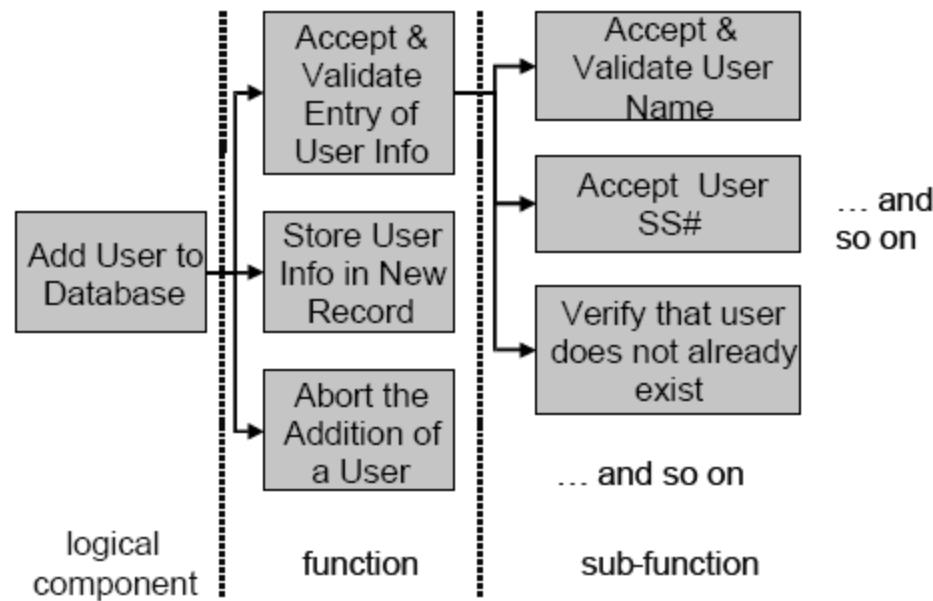


- **Bước 1:** Phân chia hệ thống thành các vùng hoạt động
- **Bước 2:** Xác định các nguyên nhân (causes), kết quả (effects)
- **Bước 3:** Chuyển nội dung ngũ nghĩa trong đặc tả thành đồ thị liên kết các cause và effects
- **Bước 4:** Chuyển đổi đồ thị thành bảng quyết định
- **Bước 5:** Thiết lập danh sách test case từ bảng quyết định. Mỗi test case tương ứng với một cột trong bảng quyết định

# Bước 1



- Phân chia hệ thống thành các vùng hoạt động
  - Phân rã các yêu cầu chức năng thành danh sách các functions hay sub-functions



# Bước 2



- **B 2.1:** Dựa vào đặc tả, xác định các causes và chỉ định mỗi causes này 1 định danh ID
  - Một cause có thể được xem như là 1 input conditions hoặc là đại diện của 1 lớp tương đương input conditions
- **B 2.2:** Dựa vào đặc tả, xác định effects hoặc sự thay đổi trạng thái của hệ thống và chỉ định mỗi effect 1 định danh ID
  - Effect có thể là output action, output condition hay là đại diện của 1 lớp tương đương output conditions

# Xác định các causes, effects



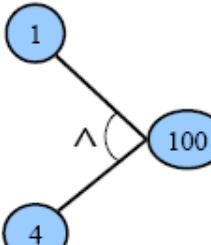
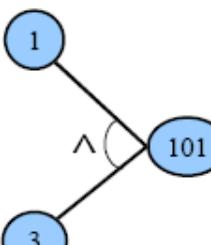
- Ví dụ: Xét đặc tả hệ thống tính phí bảo hiểm xe hơi
    - Đối với nữ < 65 tuổi, phí bảo hiểm là: 500\$
    - Đối với nam < 25 tuổi, phí bảo hiểm là: 3000\$
    - Đối với nam từ 25 đến 64, phí bảo hiểm là: 1000\$
    - Nếu tuổi từ 65 trở lên, phí bảo hiểm là: 1500\$
- Có 2 yếu tố xác định phí bảo hiểm: **giới tính và tuổi**

Causes (input conditions)	Effects (output conditions)
1. Sex is Male	100. Premium is \$1000
2. Sex is Female	101. Premium is \$3000
3. Age is <25	102. Premium is \$1500
4. Age is >=25 and < 65	103. Premium is \$500
5. Age is >= 65	

# Bước 3



- Chuyển nội dung ngũ nghĩa trong đặc tả thành đồ thị liên kết các cause và effects
  - CEG #1: Đối với nam từ 25 đến 64, phí bảo hiểm là 1000\$

CEG	Interpretation
CEG #1: 	Causes: 1. Sex is Male and (^) 4. Age is $\geq 25$ and $< 65$ Effect: 100: Premium is \$1000
CEG #2: 	Causes: 1. Sex is Male and (^) 3. Age is $< 25$ Effect: 101: Premium is \$3000

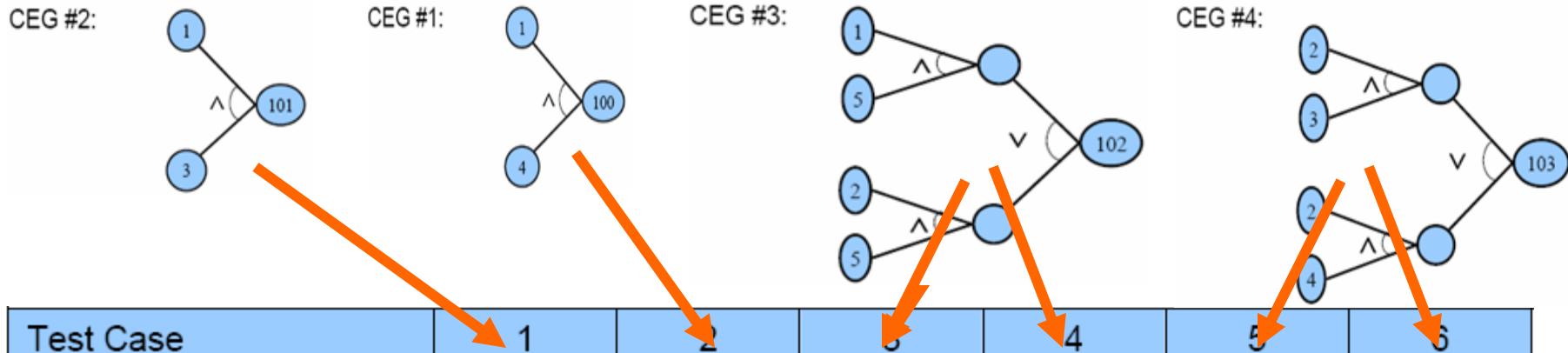
# Bước 3



- CEG #3: Nếu tuổi từ 65 trở lên, phí bảo hiểm là: 1500\$
- CEG #4: Đối với nữ < 65 tuổi, phí bảo hiểm là: 500\$

CEG #3: 	Causes: 1. Sex is Male and ( $\wedge$ ) 5. Age is $\geq 65$ or ( $\vee$ ) 2. Sex is Female and ( $\wedge$ ) 5. Age is $\geq 65$ Effect: 102: Premium is \$1500
CEG #4: 	Causes: 2. Sex is Female and ( $\wedge$ ) 3. Age is $< 25$ or ( $\vee$ ) 2. Sex is Female and ( $\wedge$ ) 4. Age is $\geq 25 \text{ and } < 65$ Effect: 103: Premium is \$500

# Bước 4: Chuyển đổi đồ thị thành Bảng quyết định



Test Case	1	2	3	4	5	6
Causes:						
1 (male)	1	1	1	0	0	0
2 (female)	0	0	0	1	1	1
3 (<25)	1	0	0	0	1	0
4 ( $\geq 25$ and $< 65$ )	0	1	0	0	0	1
5 ( $\geq 65$ )	0	0	1	1	0	0
Effects:						
100 (Premium is \$1000)	0	1	0	0	0	0
101 (Premium is \$3000)	1	0	0	0	0	0
102 (Premium is \$1500)	0	0	1	1	0	0
103 (Premium is \$500)	0	0	0	0	1	1

# Bước 5: Lập danh sách test case từ Bảng quyết định



Test Case	1	2	3	4	5	6
Causes:						
1 (male)	1	1	1	0	0	0
2 (female)	0	0	0	1	1	1
3 (<25)	1	0	0	0	1	0
4 (>=25 and < 65)	0	1	0	0	0	1
5 (>= 65)	0	0	1	1	0	0
Effects:						
100 (Premium is \$1000)	0	1	0	0	0	0
101 (Premium is \$3000)	1	0	0	0	0	0
102 (Premium is \$1500)	0	0	1	1	0	0
103 (Premium is \$500)	0	0	0	0	1	1

Test Case #	Inputs (Causes)		Expected Output (Effects) Premium
	Sex	Age	
1	Male	<25	\$3000
2	Male	>=25 and < 65	\$1000
3	Male	>= 65	\$1500
4	Female	>= 65	\$1500
5	Female	<25	\$500
6	Female	>=25 and < 65	\$500

# Bài tập 1



- Nếu bạn đi xe điện chuyến trước 9:30 sáng hoặc từ sau 4:00 chiều đến 7:30 tối (giờ cao điểm), thì bạn phải mua vé thường. Vé tiết kiệm (giá thấp hơn vé thường) có hiệu lực cho các chuyến xe từ 9:30 sáng đến 4:00 chiều và sau 7:30 tối. Tàu hoạt động từ 4:00 sáng tới 23:00 đêm
- **Thiết kế các ca kiểm thử để kiểm tra yêu cầu trên dựa vào phương pháp phân vùng tương đương và phân tích giá trị biên.**

## Bài tập 2



- TPPM “xét đơn cầm cố nhà” với đặc tả như sau: mỗi lần nhận 1 đơn xin cầm cố, phần mềm sẽ ra quyết định chấp thuận nếu 4 điều kiện sau thỏa mãn:
  - Thu nhập hàng tháng của người nộp đơn nằm trong khoảng từ 1000\$ với 83333\$
  - Số nhà xin cầm cố từ 1-5
- Dùng phương pháp phân hoạch tương đương và phân tích giá trị biên để thiết kế các trường hợp kiểm thử cho TPPM trên.

# Bài tập 3



- Viết chương trình dịch, trong đó có câu lệnh FOR, đặc tả câu lệnh FOR như sau: “Lệnh FOR chỉ chấp nhận một tham số duy nhất là biến đếm. Tên biến không được sử dụng quá hai ký tự khác rỗng. Sau ký hiệu = là cận dưới và cận trên của biến đếm. Các cận trên và cận dưới là các số nguyên dương và được đặt giữa từ khóa TO”.
- Dùng phương pháp phân hoạch tương đương, thiết kế các ca kiểm thử cho câu lệnh FOR

# Bài tập 4



- Bài toán tìm nghiệm thực cho phương trình bậc 2:
- Biết a,b,c là các số thực  $\in [-10, 100]$
- Đầu ra có thể gấp sau khi nhập bộ 3 số a,b,c và bấm nút Calculate là:
  - 1. Không phải là phương trình bậc 2.
  - 2. Phương trình vô nghiệm
  - 3. Phương trình có một nghiệm (đưa ra giá trị của nghiệm)
  - 4. Phương trình có 2 nghiệm (đưa ra giá trị của 2 nghiệm)
  - 5. Nhập sai dữ liệu
- Thiết kế các ca kiểm thử dùng phương pháp phân hoạch tương đương

*Calculator for the quadratic equation:*

$$0 = ax^2 + bx + c$$

a : :  b : :  c : :

# Bài tập 5



- Chương trình tính chi phí cho bệnh nhân dựa trên độ tuổi và giới tính có màn hình và các yêu cầu như sau:

**Calculate the Payment for the Patient**

Male       Female       Child (0 - 17 years)

Age (Years)

Payment is  euro €

Male	
Age	Payment
18-35	100 euro
36-50	120 euro
51-145	140 euro

Female	
Age	Payment
18-35	80 euro
36-50	110 euro
51-145	140 euro

Child	
Age	Payment
0-17	50 euro

# Bài tập 5



- Mô tả chức năng:
  - 1. Khởi tạo màn hình:
  - Item 2 được check mặc định ở "Male"
  - Item 3 và 5 null
  - Item 4 ở trạng thái enable (có thể click được)
  - 2. Mô tả xử lý chính:
    - Khi click vào item 4 thì xử lý như sau:
      - + Nếu là Male
        - ++ Độ tuổi từ 18 đến 35 thì nhận được 100€
        - ++ Độ tuổi từ 36 đến 50 thì nhận được 120€
        - ++ Độ tuổi từ 51 đến 145 thì nhận được 140€
        - ++ Độ tuổi khác thì hiển thị thông báo lỗi: "Xin vui lòng nhập độ tuổi chính xác"
      - + Nếu là Female
        - ++ Độ tuổi từ 18 đến 35 thì nhận được 80€
        - ++ Độ tuổi từ 36 đến 50 thì nhận được 110€
        - ++ Độ tuổi từ 51 đến 145 thì nhận được 140€
        - ++ Độ tuổi khác thì hiển thị thông báo lỗi: "Xin vui lòng nhập độ tuổi chính xác"

# Bài tập 5



- Dùng phương pháp phân hoạch tương đương và phân tích giá trị biên để xây dựng các ca kiểm thử cho chương trình trên.

# BT1 (Giải)



- Bảng phân vùng tương đương

Lịch	00:00- 3:59	4:00- 9:29	9:30- 16:00	16:01- 19:30	19:31- 22:59	23:00- 23:59
Loại vé		Vé thường	Vé tiết kiệm	Vé thường	Vé tiết kiệm	
Lớp tương đương	Ko Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Hợp lệ	Ko Hợp lệ

- Các ca kiểm thử  
– (Bảng 1)

Ca	Đầu vào	Đầu ra mong đợi
1	Xuất phát lúc 2h	Không hợp lệ
2	Xuất phát lúc 6h	Vé thường
3	Xuất phát lúc 12h	Vé tiết kiệm
4	Xuất phát lúc 18h	Vé thường
5	Xuất phát lúc 21h	Vé tiết kiệm
6	Xuất phát lúc 23h30	Không hợp lệ



- Các giá trị biên cần kiểm thử (Bảng 2)

Ca	Đầu vào	Đầu ra mong đợi
1	Xuất phát lúc 0h	Không hợp lệ
2	Xuất phát lúc 3h59	Không hợp lệ
3	Xuất phát lúc 4h	Vé thường
4	Xuất phát lúc 9h29	Vé thường
5	Xuất phát lúc 9h30	Vé tiết kiệm
6	Xuất phát lúc 16h	Vé tiết kiệm
7	Xuất phát lúc 16h01	Vé thường
8	Xuất phát lúc 19h 30	Vé thường
9	Xuất phát lúc 19h31	Vé tiết kiệm
10	Xuất phát lúc 22h59	Vé tiết kiệm
11	Xuất phát lúc 23h	Không hợp lệ

# BT1



- Kết hợp Bảng 1 &2 ta có ca kiếm thử:

Ca	Đầu vào	Đầu ra mong đợi
1	Xuất phát lúc 0h	Không hợp lệ
2	Xuất phát lúc 2h	Không hợp lệ
3	Xuất phát lúc 3h59	Không hợp lệ
4	Xuất phát lúc 4h	Vé thường
5	Xuất phát lúc 6h	Vé thường
6	Xuất phát lúc 9h29	Vé thường
7	Xuất phát lúc 9h30	Vé tiết kiệm
8	Xuất phát lúc 12h	Vé tiết kiệm
9	Xuất phát lúc 16h	Vé tiết kiệm
10	Xuất phát lúc 16h01	Vé thường
11	Xuất phát lúc 18h	Vé thường
12	Xuất phát lúc 19h30	Vé thường
13	Xuất phát lúc 19h31	Vé tiết kiệm
14	Xuất phát lúc 21h	Vé tiết kiệm
15	Xuất phát lúc 22h59	Vé tiết kiệm
16	Xuất phát lúc 23h	Không hợp lệ
17	Xuất phát lúc 23h30	Không hợp lệ

# Bài tập 2 (Giải)



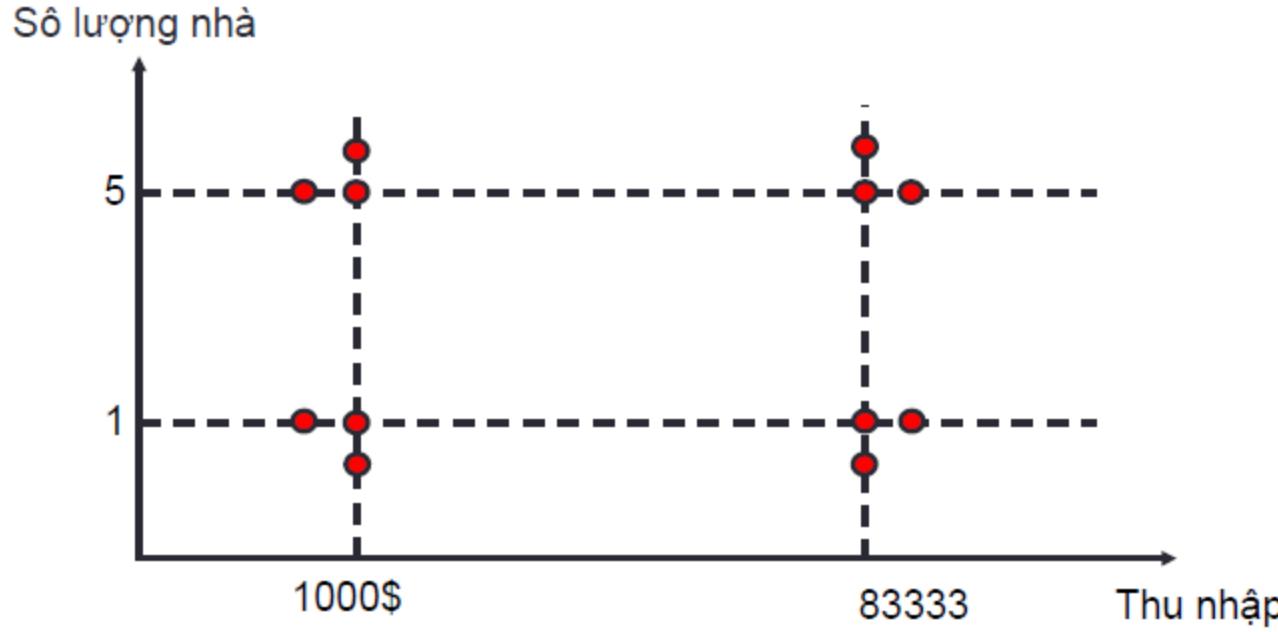
- Thu nhập hàng tháng

Đầu vào	Lớp hợp lệ	Đánh dấu	Lớp ko hợp lệ	Đánh dấu
Thu nhập hàng tháng	[1000\$ ,83333\$]	H1	<1000\$	K1
			>8333\$	K2
Số nhà cầm cố	[1,5]	H2	<1	K3
			>5	K4

- Các ca kiểm thử TC1(H1,H2), TC2(H1,K3), TC3(H1,K4), TC4(K1,H2), TC5(K2, H2)



- Các giá trị biên cần kiểm tra
  - Với Thu nhập hàng tháng {999\$, 1000\$, 83333\$, 83334\$}
  - Với số nhà cầm cố {0,1,5,6}



## BT2: Các ca kiểm thử



TC	Đầu vào		Đầu ra mong đợi
	Thu nhập	Số lượng nhà	
1	1000\$	1	Được thế chấp
2	5000\$	3	Được thế chấp
3	83333\$	1	Được thế chấp
4	1000\$	5	Được thế chấp
5	8333\$	5	Được thế chấp
6	1000\$	0	Không được thế chấp
7	8333\$	6	Không được thế chấp
8	8333\$	0	Không được thế chấp
9	8333\$	6	Không được thế chấp
10	999\$	1	Không được thế chấp
11	8334\$	1	Không được thế chấp
12	999\$	5	Không được thế chấp
13	8334\$	5	Không được thế chấp

# BT3: bảng phân vùng tương đương



Đầu vào	Hợp lệ	Đánh dấu	Ko Hợp lệ	Đánh dấu
Tên biến	1-2 k. tự	H1	Rỗng >2 k. tự	K1 K2
Số lượng biến	1	H2	0 >1	K3 K4
Cận trên	Số Nguyên dương	H3	Không phải số nguyên dương Không phải số nguyên Không phải số Rỗng	K5 K6 K7 K8
Cận dưới	Số nguyên dương	H4	Không phải số nguyên dương Không phải số nguyên Không phải số Rỗng	K9 K10 K11 K12
Toán tử gán	=	H5	K. tự khác dấu =	K13
Từ khóa To	To	H6	K. tự khác To	K14

# BT3 Test case



TC	Inputs	E. Output	Cover
1	i= 10 To 100	Vòng lặp hợp lệ	(H1,H2,H3,H4,H5,H6)
2	= 10 To 100	Không tìm thấy biến	(K1,H2,H3,H4,H5,H6) (H1,K3,H3,H4,H5,H6)
3	Bien= 10 To 100	Tên biến ko hợp lệ	(K2,H2,H3,H4,H5,H6)
4	i,j=10 To 100	Thừa biến	(H1,K4,H3,H4,H5,H6)
5	i=-10 To 100	Sai cận dưới	(H1,H2,K5,H4,H5,H6)
	i= 1.5 To 100	Sai cận dưới	(H1,H2,K6,H4, H5,H6)
6	i= a To 100	Sai cận dưới	(H1,H2,K7,H4,H5,H6)
7	i= To 100	Không tìm thấy cận dưới	(H1,H2,K8,H4,H5,H6)
8	i= 10 To -100	Sai cận trên	(H1,H2,H3,K9,H5,H6)
9	i=10 To 10.25	Sai cận trên	(H1,H2,H3,K10,H5,H6)
10	i =10 To a	Sai cận trên	(H1,H2,H3,K11,H5,H6)
11	i= 10 To	Không tìm thấy cận trên	(H1,H2,H3,K12,H5,H6)
13	i <> 1 To 100	Sai cú pháp lệnh	(H1,H2,H3,H4,K13,H6)
14	i=1 Tuo 100	Sai từ khóa	(H1,H2,H3,H4,H5,K14)

# BT4 (Giải)



- **Bảng phân hoạch tương đương**

Đầu vào	Vùng hợp lệ	Đánh dấu	Vùng ko hợp lệ	Đánh dấu
A	[-10, 100]	H1	<-10 >100 Không phải số =0 Để trống	K1 K2 K3 K4 K5
B	[-10, 100]	H2	<-10 >100 Không phải số Để trống	K6 K7 K8 K9
C	[-10, 100]	H3	<-10 >100 Không phải số Để trống	K10 K11 K12 K13

# Các ca kiểm thử



- Test case

TC	inputs	E. output	Cover
	3 số a,b,c	Delta	
1	a=2, b=5, c= 2	Delta >0 Delta =9 Pt có 2 nghiệm {-1/2,-2}	H1,H2,H3
2	a=4, b=-4, c= 1	Delta=0 Delta=0 Pt có 1 nghiệm {1/2}	H1,H2,H3
3	a=1, b=-7, c= 16	Delta<0 Pt vô nghiệm	H1,H2,H3
4	a=-20, b=5, c= 2	Nhập sai dữ liệu	K1,H2,H3
5	a=200, b=5, c= 2	Nhập sai dữ liệu	K2,H2,H3
6	a= abc, b=5, c=2	Nhập sai dữ liệu	K3,H2,H3
7	a=0, b=5, c=2	Không phải pt bậc 2	K4,H2,H3
8	Đè trắng, b=5, c=2	Nhập sai dữ liệu	K5,H2,H3
9	a= 2, b=-50, c=2	Nhập sai dữ liệu	H1, K6,H3

# Các ca kiểm thử (tiếp)



- Test case

TC	Inputs	E. output	Cover
	3 số a,b, c		
10	a= 2, b=500, c=2	Nhập sai dữ liệu	H1,K7,H3
11	a= 2, Nhập abcde, c=2	Nhập sai dữ liệu	H1,K8, H3
12	a= 2, để trống, c=2	Nhập sai dữ liệu	H1,K9,H3
13	a= 2, b=5, c=-20	Nhập sai dữ liệu	H1,H2,K10
14	a= 2, b=5, c=200	Nhập sai dữ liệu	H1,H2,K11
15	a= 2, b=5, Nhập abcde	Nhập sai dữ liệu	H1,H2,,K12
16	a= 2, b=5 , Để trống	Nhập sai dữ liệu	H1,H2,K13

# Bài tập 1 (Bảng quyết định)



- Nếu bạn có thẻ đường sắt "over 60s" thì được giảm giá 34% trên tất cả các vé bạn mua.
- Khi bạn đi cùng với trẻ em (dưới 16 tuổi), thì bạn sẽ được giảm 50% nếu bạn có thẻ "family rail card", trong trường hợp ko có thẻ bạn chỉ được giảm 10%
- Bạn chỉ được sử dụng 1 hình thức khuyến mại trong 1 giao dịch
- Hãy viết bảng quyết định liệt kê toàn bộ các kết hợp loại thẻ và kết quả giảm giá. Và viết test case từ bảng quyết định này

# Bài tập 2 (Bảng quyết định)



- Một chương trình phân loại kết quả học của sinh viên dựa trên tổng điểm. Biết tổng điểm của sinh viên (tối đa là 100) trong một kỳ thi thành phần cộng điểm thi.
- Trong đó điểm thi tối đa là 75 điểm, điểm thành phần tối đa là 25 điểm
- Kết quả được phân loại như sau

Tổng điểm TD	Kết quả
--------------	---------

- |                   |   |
|-------------------|---|
| $TD > 70$         | A |
| $50 < TD \leq 70$ | B |
| $30 < TD \leq 50$ | C |
| $TD \leq 30$      | D |

- Xây dựng các ca kiểm thử dựa trên:
  - 1. PP phân hoạch tương đương
  - 2. PP phân tích giá trị biên
  - 3. PP bảng hỗ trợ quyết định

# BT1 (Giải)



- Lập bảng quyết định

		L1	L2	L3	L4	L5	L6	L7	L8
Điều kiện	<b>Thẻ over 60s</b>	Y	Y	Y	Y	N	N	N	N
	<b>Thẻ family</b>	Y	Y	N	N	Y	Y	N	N
	<b>Đi cùng trẻ em &lt;16t</b>	Y	N	Y	N	Y	N	Y	N
Hành động	<b>Giám giá 34%</b>		Y	Y	Y				
	<b>Giám giá 50%</b>	Y				Y			
	<b>Giám giá 10%</b>							Y	
	<b>Không giảm</b>						Y		Y

Rút gọn L2,4 và L6,8

# BT1



- Xây dựng testcase (có tất cả 6 testcase)

TC	Inputs	Expected outputs
1	Gia đình Smiths có thẻ over 60s, có thẻ family rail card, đi cùng Oliver (10 tuổi)	Được giảm 50%
2	Gia đình Smiths có thẻ over 60s, không đi cùng trẻ em	Được giảm 34%
3	Gia đình Smiths có thẻ over 60s, đi cùng Oliver (10 tuổi)	Được giảm 34%
4	Gia đình Smiths ko có thẻ over 60s, có thẻ family raid card, đi cùng Oliver (10t)	Được giảm 50%
5	Ông Smith ko có thẻ over 60s, ko có thẻ family raid card, đi cùng Oliver (10t)	Được giảm 10%
6	Ông Smith đi một mình, ko có thẻ nào	Không được giảm

# BT2 (Giải)



## • A. Bảng phân hoạch tương đương

Đầu vào	Điểm thi	Điểm thành phần	Điểm tổng	Kết quả
Vùng hợp lệ	[0,75] (H1)	[0,25] (H2)	TĐ>70	A
			50< TĐ <= 70	B
			30 < TĐ <= 50	C
			TĐ <= 30	D
Vùng không hợp lệ	<0 (K1)	<0 (K3)		
	>75 (K2)	>25 (K4)		



- Các ca kiểm thử

TC	Inputs Điểm thi	Điểm TP	E.Output Điểm tổng	Cover
1	60	15	75 A	H1 H2
2	30	20	60B	H1 H2
3	20	20	35C	H1 H2
4	10	20	20 D	H1 H2
5	60	-10	Không hợp lệ	H1 K3
6	60	30	Không hợp lệ	H1 K4
7	60	20	Không hợp lệ	K1 H2
8	-60	20	Không hợp lệ	K2 H2



- B Phân tích các giá trị biên

TC	Inputs		E. Output	TC	Inputs		E. Output
	Điểm thi	Điểm TP			Điểm thi	Điểm TP	
1	-1	0	Ko hợp lệ	13	0	1	1 D
2	-1	25	Ko hợp lệ	14	13	16	29 D
3	75	26	Ko hợp lệ	15	14	16	30 D
4	76	25	Ko hợp lệ	16	15	16	31 C
5	75	-1	Ko hợp lệ	17	20	29	49 C
6	76	0	Ko hợp lệ	18	21	29	50 C
7	0	26	Ko hợp lệ	19	22	29	51 B
8	0	-1	Ko hợp lệ	20	49	20	69 B
9	0	0	0 D	21	50	20	70 B
10	0	25	25 D	22	51	20	71 A
11	75	0	75 A	23	75	24	99 A
12	75	25	100 A				