Kiểm thử bằng phân tích giá trị biên

(Boundary Value Analysis \_ **BVA**)

**1/. Giới thiệu:**

*Tham khảo từ các nguồn sau:*

<https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html>

<https://www.javatpoint.com/boundary-value-analysis-in-black-box-testing>

<https://www.softwaretestingmaterial.com/boundary-value-analysis-testing-technique/>

<https://viblo.asia/p/tim-hieu-ve-ky-thuat-phan-tich-gia-tri-bien-va-phan-vung-tuong-duong-trong-kiem-thu-hop-den-bWrZnBJrZxw>

Kiểm thử bằng phân tích giá trị biên là một trong những kỹ thuật được áp dụng phổ biển nhất trong kiểm thử . Chúng ta sẽ coi một chương trình là một hàm toán học với đầu vào của chương trình tương ứng với các tham số của hàm và đầu ra của chương trình là giá trị trả về của hàm. Vì hàm toán học là ánh xạ từ miền xác định của hàm đến miền giá trị của hàm. Chúng ta sẽ tập trung vào xem xét các giá trị biên và cạnh biên của hai miền đầu vào và đầu ra (input, output) của hàm để xây dựng các Test case.

**2/. Phân tích giá trị biên:**

Chủ yếu là xem xét các giá trị tập trung vào miền xác định, và dựa trên các miền giá trị để xác định các Test case. Lý do là các lỗi thường xảy ra tại giá trị biên và gần các giá trị biên này. Các chương trình viết bằng ngôn ngữ không có kiểm tra kiểu mạnh càng cần phải kiểm thử giá trị biên như Javascript, php, Visual Basic.

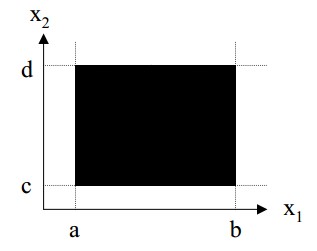
**Giá trị biên**

Giá trị biên có thể xét trên 1 hoặc nhiều biến, số lượng giá trị biên sẽ tăng theo các biến.

Giả sử y = f(x1,x2) với x1,x2,y ∈ N là một hàm toán học của một chương trình. Khi đó thông thường x1 và x2 có miền xác định thể hiện bằng các biến.

**Vd:** a ≤ x1 ≤ b và c ≤ x2 ≤ d

trong đó a,b,c,d là các hằng số nào đó. Hình sau thể hiện miền xác định của hai biến này.



Với dữ liệu được nhập vào, do lỗi lập trình hoặc đặc tả làm các biểu thức điều kiện không chính xác. Chẳng hạn đúng ra phải là dấu **<=** nhưng người lập trình hoặc đặc tả lại chỉ viết **<** hoặc ngược lại. Kiểm thử với các giá trị biên giúp chúng ta phát hiện các lỗi này. Trong nhiều trường hợp các biên này là ẩn, không được viết rõ ra trong yêu cầu nên lập trình viên dễ mắc lỗi không kiểm tra các giá trị đầu vào, hoặc không kiểm tra kết quả có hợp lệ không trước khi trả về.

Ví dụ: Xét đoạn chương trình sau, biên ở đâu ?

**if** (sokm<=5.0)

stpt=sokm \* 10000.0;

**else** **if** (sokm>5.0 && sokm<30.0)

stpt=(5 \* 10000.0) + ((sokm-5)\*8500.0);

**else** **if** (sokm>30.0 && sokm<100.0)

stpt=(5 \* 10000.0) + (25\*8500.0) + ((sokm-30)\*6800.0);

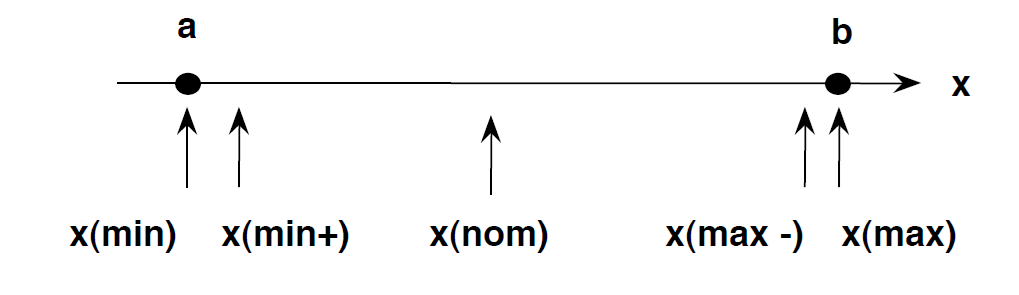
**else** **if** (sokm>100.0 && sokm<=150.0)

stpt=(50000.0) + (25\*8500.0) + (70\*6800.0) + ((sokm-100)\*5500.0);

**else**

stpt=5000.0 \* sokm;

Để tăng khả năng phát hiện lỗi, kiểm thử giá trị biên thường xác định 05 giá trị: Max (cực đại), Min (cực tiểu), Min+, Max- (các giá trị cận biên), Nom (giá trị ở giữa miền xác định đại diện cho giá trị thông thường).



(Min, Min+, Nom, Max-, Max)

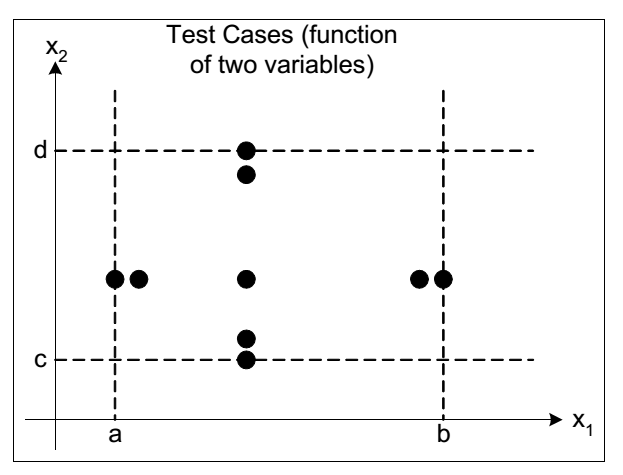
Tổng quát hóa kiểm thử bằng phân tích giá trị biên cho hàm n biến số và mỗi biến có các giá trị biên và cận biên khác nhau ta có thể dễ thấy sẽ có (1+4n) Test case vì xuất phát từ một Test case gồm các giá trị trung bình của các biến, ta có 04 giá trị biên và cận biên: Min, Min+, Max, và Max−. Với n biến ta có F = (4n + 1) là số Test case

Do đó tổng số Test case là **4n + 1**. Tuy nhiên tùy theo miền xác định của biến mà số lượng này thực tế có thể ít hơn. Ví dụ biến nguyên thuộc khoảng [1,2] thì không có cận biên và không có giá trị ở giữa. Hình sau minh họa (4n + 1) Test case = 9 Test case cho hàm 2 biến.

Số Test case T = {(x1Nom,x2Nom) giữa, (x1Min,x2Nom),(x1Min+,x2Nom),

(x1Max−,x2Nom),(x1Max,x2Nom) dòng, (x1Nom,x2Min),(x1Nom,x2Min+),

(x1Nom,x2Max−),(x1Nom,x2Max) cột}



**3/. Một số dạng kiểm thử giá trị biên**

**3.1/. Kiểm thử giá trị biên bình thường:** (trường hợp đơn giản có 1 biến)

Là kiểm thử với bộ giá trị: (Min, Min+, Nom, Max-, Max)

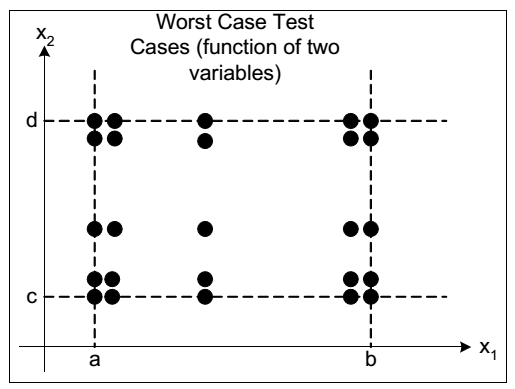
Kiểm thử trong vùng dữ liệu hợp lệ (Valid)

**3.2/. Kiểm thử giá trị biên tổ hợp** (trường hợp có >= 2 biến)

Kiểm vùng trong dữ liệu hợp lệ (Valid).

Ở trên chúng ta chỉ tạo các Test case với giá trị biên và cận biên cho từng biến. Nếu chúng ta mở rộng với hai hoặc với tất cả các biến đều được đẩy đến giá trị biên và cận biên thì chúng ta sẽ tạo ra được các Test case giá trị biên tổ hợp. Tức là từ bộ giá trị 5 phần tử (Min, Min+, Nom, Max− và Max) để tạo ra các Test case. Với n=2 biến ta có F = 52 = 25

Với hai biến hình sau minh họa các Test case này.



*Các Test case biên tổ hợp của hàm hai biến.*

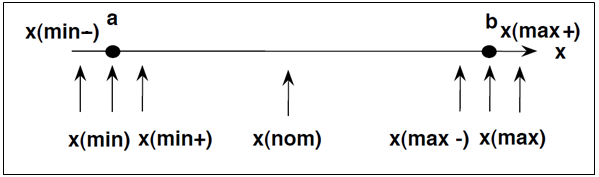
Có thể thấy cách tổ hợp các biên và cận biên này sẽ kiểm tra kỹ hơn kiểm thử giá trị biên thông thường. Tuy nhiên số Test case theo cách này tăng lên đáng kể, so với 4n+1 so với kiểm thử biên thông thường.

**3.3/. Kiểm thử giá trị biên mạnh:**

Kiểm vùng hợp lệ (Valid) và cả vùng không hợp lệ (Invalid)

Kiểm thử giá trị biên mạnh là mở rộng của kiểm thử giá trị biên bằng việc bổ sung các giá trị cận biên ở bên ngoài miền xác định. Ngoài 5 giá trị biên đã nêu ở phần trước, chúng ta sẽ lấy thêm các giá trị cận biên ở ngoài miền xác định là (Max+ và Min−). Các Test case này sẽ giúp ta kiểm tra chương trình với dữ liệu không hợp lệ, nằm ngoài khoảng mong đợi. Ví dụ khi nhập ngày 32/5/2013 chương trình cần có thông báo lỗi thích hợp.

Các Test case mạnh này giúp chúng ta kiểm tra xem chương trình có xử lý các ngoại lệ hay kiểm tra biến đầu vào hay không ?.

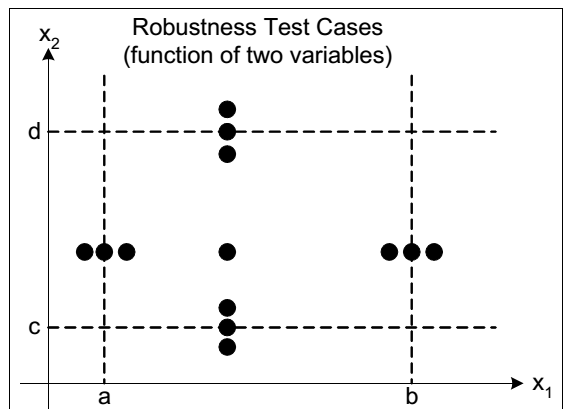


Là kiểm thử với bộ 07 giá trị: (Min-,Min, Min+, Nom, Max-, Max, Max+)

Kiểm thử cho vùng dữ liệu hợp lệ (Valid) và không hợp lệ (Invalid)

Ta có 6 giá trị cho mỗi biến, cùng 1 giá trị Nom

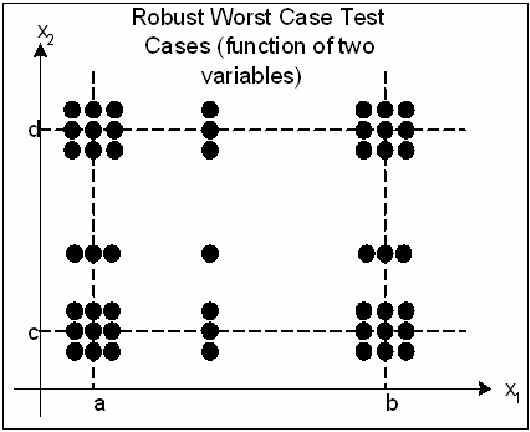
Với n biến ta có : F = (6n + 1)



Các Test case cho kiểm thử giá trị biên mạnh cho hàm hai biến.

Ta có thể mở rộng kiểm thử biên tổ hợp với bộ 7 giá trị của kiểm thử giá trị biên mạnh (Min-, Min, Min+, Nom, Max-, Max, Max+).

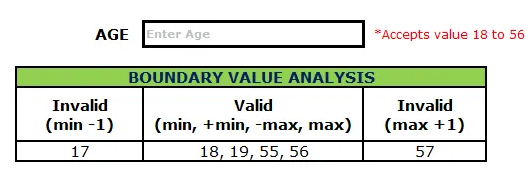
Với n=2 biến ta có : F = 72 = 49



Trường hợp này, Chúng ta sẽ kiểm tra được kỹ hơn nhưng cũng mất nhiều công sức, thời gian hơn, đến 7n = 72 = 49 Test case.

**Vd 1:** kiểm thử giá trị biên với 1 biến.

Chúng ta cần kiểm thử cho tuổi nhập vào có giá trị từ 18 – 56



Kiểm thử giá trị biên bình thường, tuổi gồm các giá trị sau : (5)

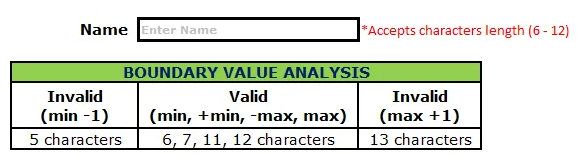
18, 19, 35, 55, 56. *(cần lấy giá trị nom 35)*

Kiểm thử giá trị biên mạnh, tuổi gồm các giá trị sau : (7)

17, 18, 19, 35, 55, 56, 57.

**Vd 2 :** kiểm thử giá trị biên với 1 biến.

Chúng ta cần kiểm thử cho tên nhập vào có độ dài từ 6 – 12



Kiểm thử giá trị biên bình thường, tên có độ dài sau : (5)

6, 7, 9, 11, 12. *(cần lấy giá trị nom 9)*

Kiểm thử giá trị biên mạnh, tên có độ dài sau : (7)

5, 6, 7, 9, 11, 12, 13.

**4/. Kiểm thử với các giá trị đặc biệt :**

Kiểm thử các giá trị đặc biệt cũng là một phương pháp phổ biến. Đây cũng là phương pháp trực quan nhất và không theo một khuôn dạng cụ thể nào. Dựa trên hiểu biết về bài toán và miền ứng dụng kết hợp với kinh nghiệm cá nhân, người kiểm thử đưa ra các giá trị kiểm thử. Do đó không có hướng dẫn cụ thể nào cho phương pháp này. Mức độ hiệu quả của phương pháp này phụ thuộc nhiều vào khả năng của người kiểm thử. Trên thực tế các đơn vị phát triển phần mềm vẫn áp dụng phương pháp này, vì nhiều khi nó giúp phát hiện lỗi nhanh, không tốn nhiều công sức.

**Vd:** dựa trên hiểu biết về số ngày của các tháng trong một năm chúng ta sẽ kiểm thử các ngày 28 tháng 2 và 29 tháng 2 ở cả năm nhuận và năm thường. Số lượng Test case sẽ ít hơn nhiều so với kiểm thử giá trị biên và cũng hiệu quả trong việc kiểm tra lỗi. Đây cũng là tính chất ’thủ công’ của nghề kiểm thử phần mềm - ’nghệ thuật’ và có kinh nghiệm sẽ làm tốt hơn.

**Vd 3:** Bài toán Tam giác (Triangle)

Chương trình yêu cầu nhập 3 cạnh a, b, c của một tam giác, với a, b, c là số nguyên có giá trị trong khoảng [1 – 200]. Kết quả của chương trình là:

* + Tam giác đều (Equilateral)
  + Tam giác cân (Isosceles)
  + Tam giác thường (Scalene)
  + Không tạo thành tam giác (Not a Triangle)

Cận dưới (Min) của cạnh tam giác là 1 và cận trên (Max) là 200.

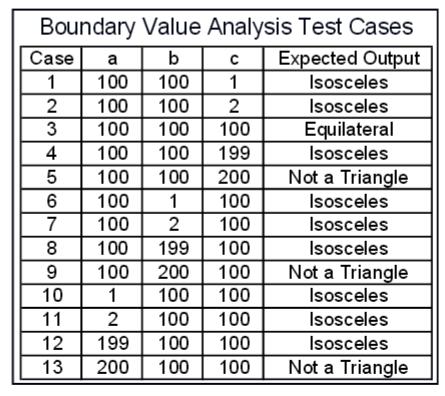
Ta có (4n + 1) = 12 + 1 = 13 Test case với T/h kiểm thử giá trị biên bình thường:

(Min = 1; Min + = 2; Nom = 100; Max - = 199; Max = 200)

Bài toán phải thoả 6 điều kiện sau:

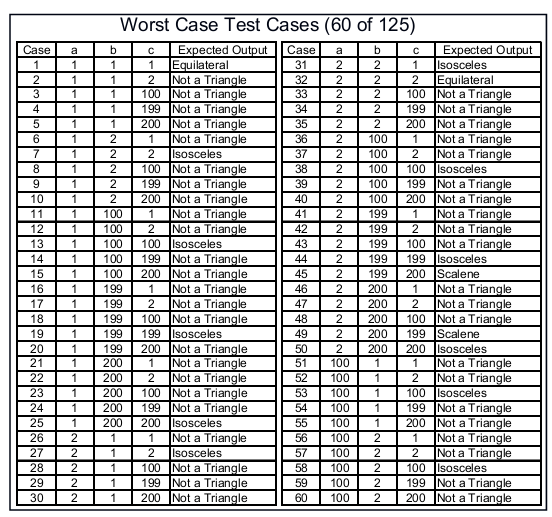
* C1. 1 a 200.
* C2. 1 b 200.
* C3. 1 c 200.
* C4. a < b + c.
* C5. b < a + c.
* C6. c < a + b.

**Kiểm thử giá trị biên bình thường:**



Ta có (53) = 125 Test case với T/h kiểm thử giá trị biên tổ hợp. Hình sau chỉ minh hoạ 60 trong 125 Test case ,..

**Kiểm thử giá trị biên tổ hợp:**



**Vd 6: Bài toán Next Date:**

Chương trình nhập vào 1 date (ngày/tháng/năm), cho biết ngày kế tiếp là ngày nào?

Bài toán NextDate là một hàm có ba biến biểu diễn ngày, tháng và năm là (day, month  
và year). Hàm này trả về ngày kế tiếp của ngày nhập vào.

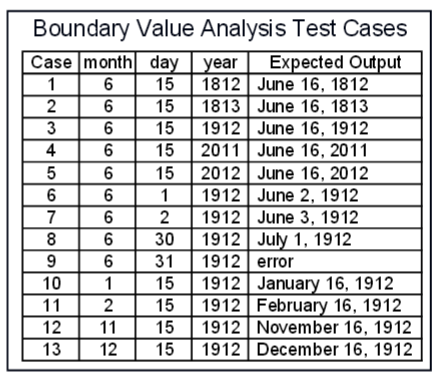
Các biến day, month, year có các giá trị số thỏa mãn các ràng buộc (miền giá trị):

* 1 *≤* day *≤* 31.
* 1 *≤* month *≤* 12.
* 1812 *≤* year *≤* 2012.

|  |  |  |
| --- | --- | --- |
| **Tháng (Month)** | **Ngày (Day)** | **Năm (Year)** |
| Min = 1  Min+ = 2  Nom = 6  Max- = 11  Max = 12 | Min = 1  Min+ = 2  Nom = 15  Max- = 30  Max = 31 | Min = 1812  Min+ = 1813  Nom = 1950  Max- = 2011  Max = 2012 |

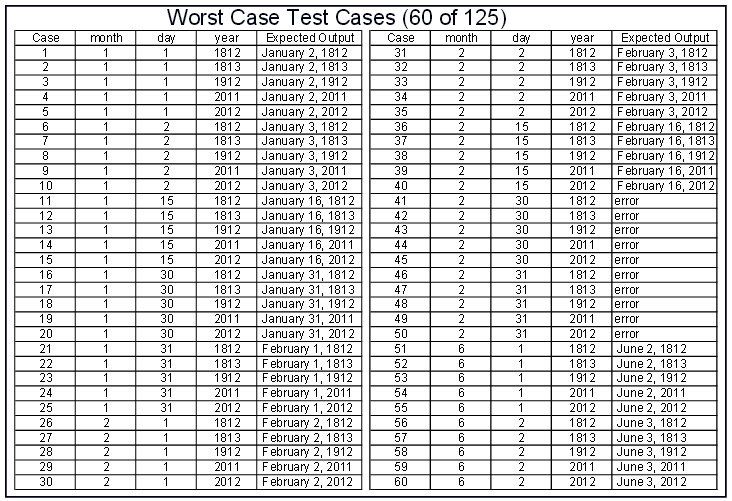
Ta có (4n + 1) = 12 + 1 = 13 Test case với T/h kiểm thử giá trị biên bình thường:

**Kiểm thử giá trị biên bình thường :**



Ta có (53) = 125 Test case với T/h kiểm thử giá trị biên tổ hợp. Hình sau chỉ minh hoạ 60 trong 125 Test case ,..

**Kiểm thử giá trị biên tổ hợp :**



**Câu hỏi:**

Kiểm thử giá trị biên tổ hợp mạnh thì kiểm thêm bộ giá trị nào ?

**A :** Kiểm thêm các giá trị Invalid cho các biến ngày, tháng, năm,…

**Kinh nghiệm áp dụng**

Kiểm thử giá trị biên là phương pháp thô nhất trong các phương pháp kiểm thử. Phương pháp này nên áp dụng cho các hàm có các biến đầu vào độc lập, không phụ thuộc vào nhau. Chúng ta giả định rằng một khiếm khuyết trong chương trình sẽ gây ra lỗi ngay thì chúng ta sử dụng kiểm thử biên thông thường. Trái lại khi lỗi thường xuất hiện khi có hơn một khiếm khuyết trong chương trình thì chúng ta cần sử dụng kiểm thử giá trị biên mạnh. Các phương pháp này đều đơn giản, dễ dàng tự động hóa việc sinh và chạy các Test case này.

**Kết luận:** Cần kết hợp nhiều kỹ thuật kiểm thử để tìm thấy lỗi hiệu quả nhất.

**Xem các mẫu Test Case**

**Bài tập nhà:**

**1/.** Hệ thống bán vé du lịch:

* trẻ em dưới 03 tuổi được phép để đi du lịch miễn phí
* người dưới 18 tuổi cũng như người già lớn tuổi hơn 64 trả $10
* người lớn khác cần phải trả $20.

**2/.** Một tài khoản tiết kiệm trong ngân hàng áp dụng tỷ lệ lãi suất khác nhau tùy thuộc vào số dư trong tài khoản. Để kiểm tra phần mềm tính toán lãi suất, chúng ta có phạm vi của các giá trị tài khoản để tính lãi suất (/ 1 năm).

* 3% lãi suất cho số dư trong tài khoản từ: >=$ 1 đến $ 100
* 4% lãi suất cho số dư trong tài khoản từ: >$ 100 đến <= $ 1000
* 7% lãi suất cho số dư trong tài khoản từ: >$ 1000 trở lên.

**3/.** Chương trình tính thuế thu nhập tính thuế dựa vào thu nhập hàng năm như sau:

* Thu nhập [10, 1000] phải nộp 2% thu nhập
* Thu nhập (1000 – 50000] phải nộp 5% thu nhập
* Thu nhập (50000 – 500000] phải nộp 10% thu nhập
* Thu nhập (500000 – 2000000] phải nộp 15% thu nhập
* Thu nhập >2000000 phải nộp 25% thu nhập

**Yêu cầu:** Kiểm thử giá trị biên tổ hợp mạnh