

Kỹ Thuật Phân Tích Giá Trị Biên

Boundary Value Analysis – BVA
Black-box Testing

Người trình bày:



Đoàn Thị Kim Nhung



BA Team



BATIZENS

Mục Lục

01

Người đề xuất
Proposer

02

Ý nghĩa
Meaning

03

Cách áp dụng
How to apply

04

Cách tính Test Case
Test Case Calculation

05

Ví dụ minh họa
Example

06

Ưu nhược điểm
Advantages & Disadvantages



BATIZENS



Từ ngữ viết tắt và định nghĩa

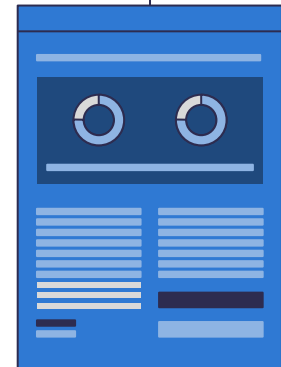
#	Từ ngữ viết tắt	Định nghĩa
1	BVA	Boundary Value Analysis
2	TC	Test Case
3	TCs	Test Cases
4		
5		
6		
7		
8		





01

Người đề xuất



Người đề xuất

Kỹ thuật Phân tích giá trị biên (Boundary Value Analysis - BVA) là một trong những kỹ thuật kiểm thử phổ biến được đề xuất bởi **Boris Beizer**, một nhà nghiên cứu và chuyên gia trong lĩnh vực kiểm thử phần mềm.

Boris Beizer đã đóng góp đáng kể vào lĩnh vực kiểm thử phần mềm thông qua các cuốn sách và công trình nghiên cứu của mình. Nổi bật nhất là cuốn “*Software Testing Techniques*” (1983).

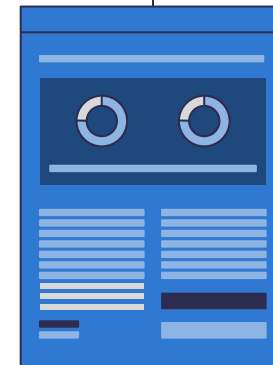


Boris Beizer (1934 – 2016)



02

Ý nghĩa





Ý nghĩa

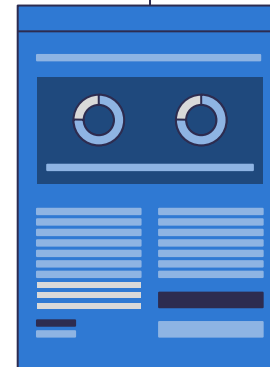
#	Ý nghĩa	Mô tả
1	Phát hiện lỗi tại các biên	BVA tập trung vào việc kiểm tra các giá trị ở biên của phạm vi đầu vào, nơi các lỗi thường xuất hiện. Bằng cách này, nó giúp phát hiện các lỗi liên quan đến xử lý biên, kiểm soát biên và kiểm tra biên.
2	Tăng cường độ bao phủ kiểm thử	Kiểm tra các trường hợp cận biên, cận biên kỳ vọng và nằm bên trên/bên dưới biên, để đảm bảo rằng các giá trị biên được kiểm tra một cách toàn diện.
3	Tăng độ tin cậy và chất lượng của phần mềm	
4	Tiết kiệm thời gian và công sức	Thay vì kiểm tra tất cả các giá trị trong một phạm vi, BVA tập trung vào các giá trị quan trọng nhất tại các biên.
5	Dễ dàng triển khai và thực hiện	Có thể triển khai một cách dễ dàng và thực hiện một cách hiệu quả, đặc biệt là trong các trường hợp phạm vi đầu vào có số lượng lớn giá trị.





03

Cách áp dụng

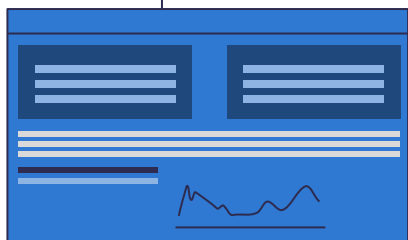




Cách áp dụng

#	Cách áp dụng	Mô tả
1	Khi có các biên quan trọng	Khi phần mềm hoạt động dựa trên các phạm vi hoặc giới hạn, việc kiểm tra các giá trị biên là rất quan trọng. Trong trường hợp này, BVA được sử dụng để kiểm tra và xác minh tính chính xác của xử lý biên.
2	Khi phạm vi đầu vào lớn	Trong các trường hợp phạm vi đầu vào lớn, việc kiểm tra tất cả các giá trị đầu vào có thể trở nên không khả thi. BVA giúp tối ưu hóa quá trình kiểm thử bằng cách tập trung vào các giá trị quan trọng nhất ở biên của phạm vi. Điều này cũng đảm bảo rằng các trường hợp biên đã được kiểm tra một cách toàn diện.
3	Khi tài nguyên hạn chế	Khi có hạn chế về thời gian, nguồn lực và kinh phí cho quá trình kiểm thử, BVA là một phương pháp hiệu quả để tối ưu hóa quá trình kiểm thử.
4	Khi kiểm thử mã nguồn mở hoặc thư viện của bên thứ ba	Trong các dự án sử dụng mã nguồn mở hoặc thư viện của bên thứ ba, việc kiểm thử các giá trị biên quan trọng có thể giúp đảm bảo tính ổn định và tin cậy của các tính năng được tích hợp.





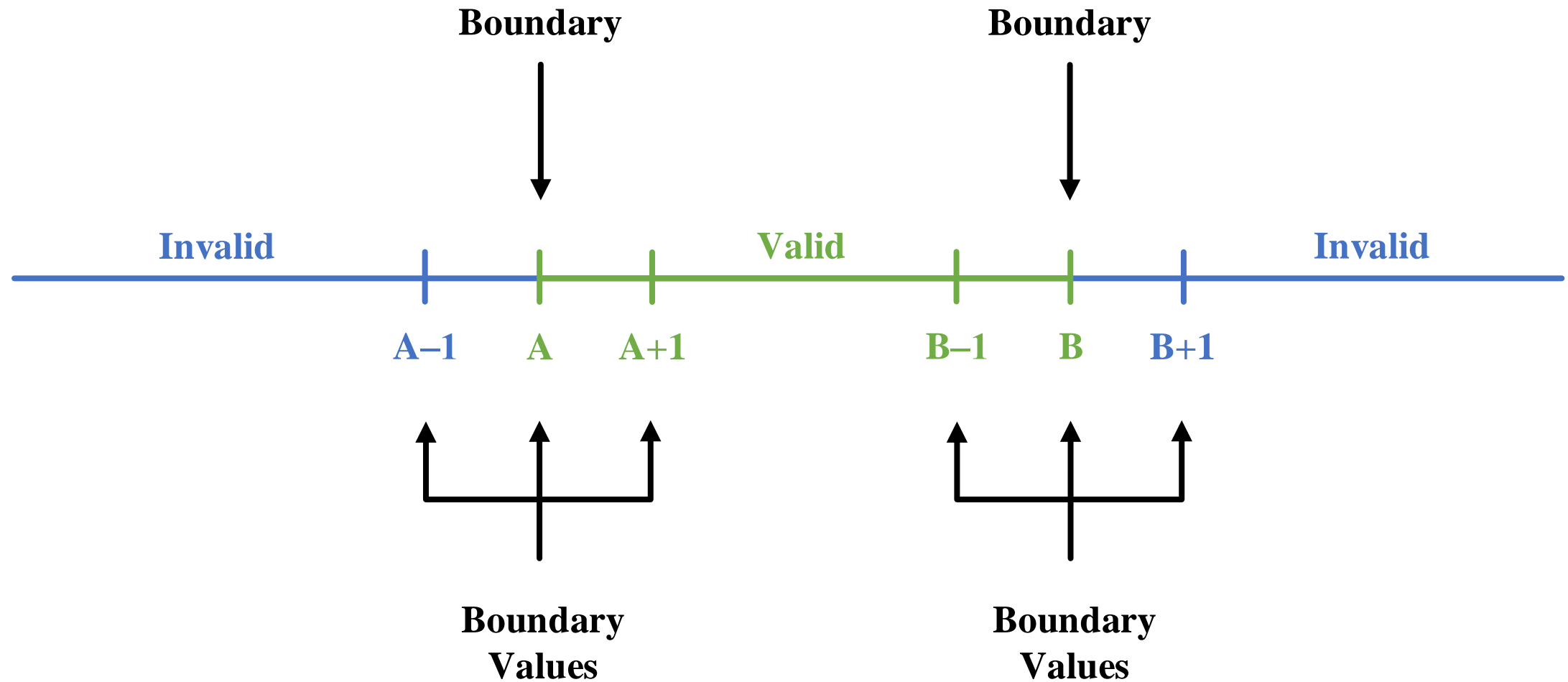
04

Cách tính số lượng Test Case



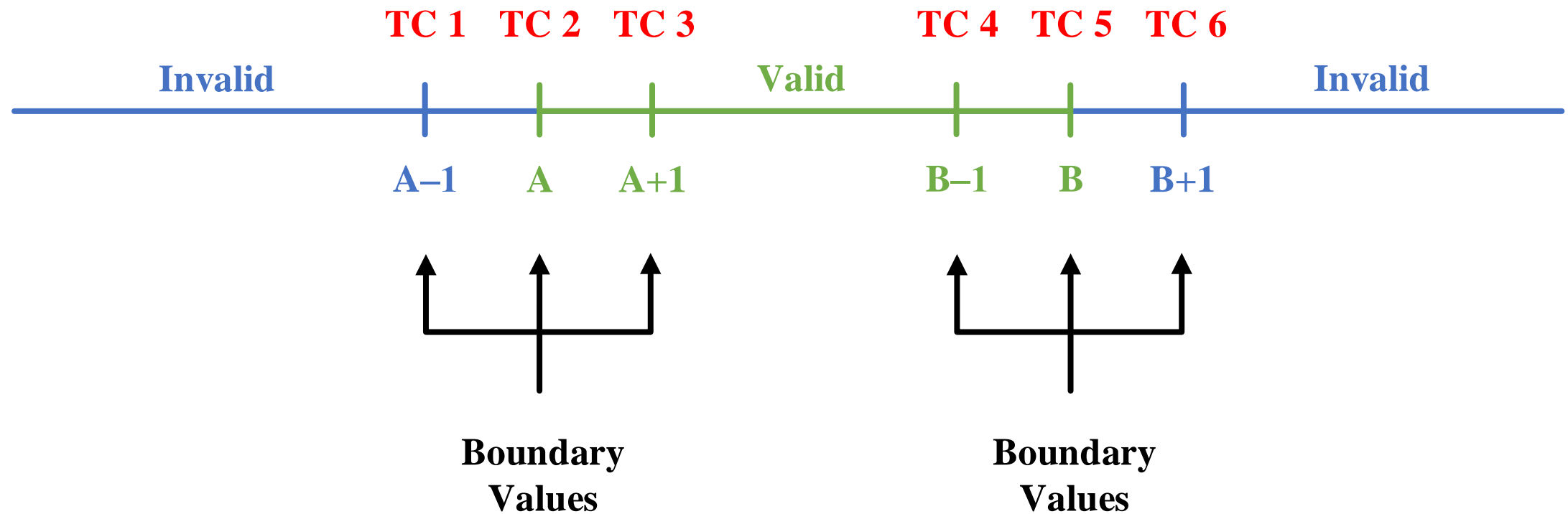


Cách tính Test Case





Cách tính Test Case



Cách tính Test Case

TH 1: 1 biên có tổng 1 giá trị biên (biên gốc)



$$1 * p$$

- p: Tổng số biên mà mỗi biên có tổng 1 giá trị biên.

TH 2: 1 biên có tổng 2 giá trị biên (biên gốc và biên +1 hoặc biên -1)



$$2 * q$$

- q: Tổng số biên mà mỗi biên có tổng 2 giá trị biên.

TH 3: 1 biên có tổng 3 giá trị biên (biên gốc, biên +1 và biên -1)



$$3 * n$$

- n: Tổng số biên mà mỗi biên có tổng 3 giá trị biên.

Cách tính Test Case

TH 1: 1 biên có tổng 1 giá trị biên (biên gốc)

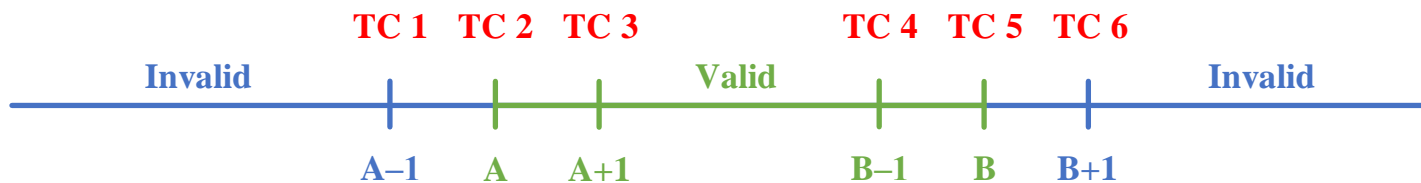


TH 2: 1 biên có tổng 2 giá trị biên (biên gốc và biên +1 hoặc biên -1)



$$1 * p + 2 * q + 3 * n$$

TH 3: 1 biên có tổng 3 giá trị biên (biên gốc, biên +1 và biên -1)



Cách tính Test Case

TH 1: 1 biên có tổng 1 giá trị biên (biên gốc)



TH 2: 1 biên có tổng 2 giá trị biên (biên gốc và biên +1 hoặc biên -1)



TH 3: 1 biên có tổng 3 giá trị biên (biên gốc, biên +1 và biên -1)



$$\sum_{i=1}^n m_i$$

- i (1...n): Biên thứ i.
- m: Tổng số biên của biên thứ i.



Cách tính Test Case

Công thức tổng quát tính số lượng Test Case theo kỹ thuật BVA.

Cách 1

$$Total\ TCs = 1 * p + 2 * q + 3 * n$$

Cách 2

$$Total\ TCs = \sum_{i=1}^n m_i$$



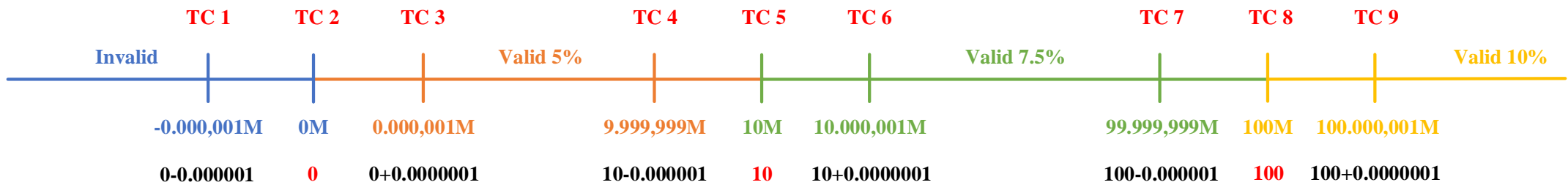
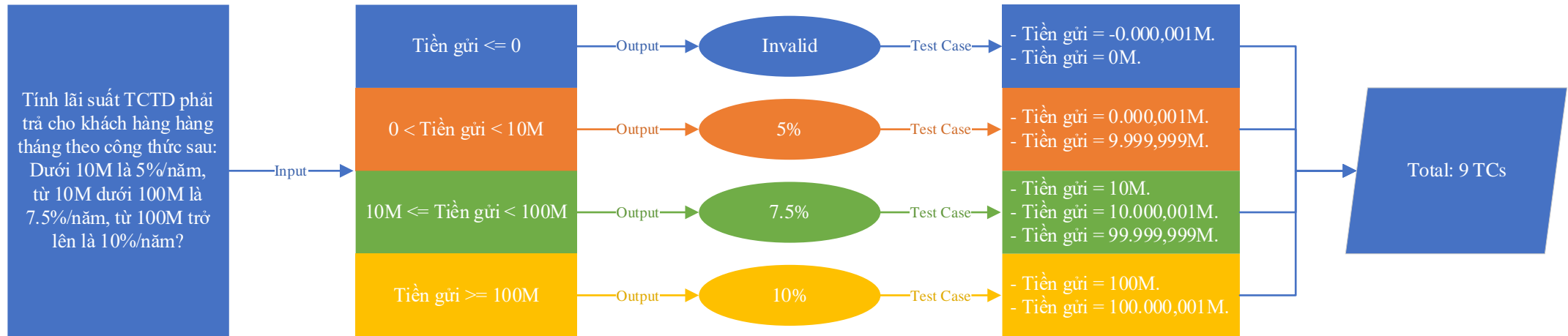


05

Ví dụ minh họa về BVA

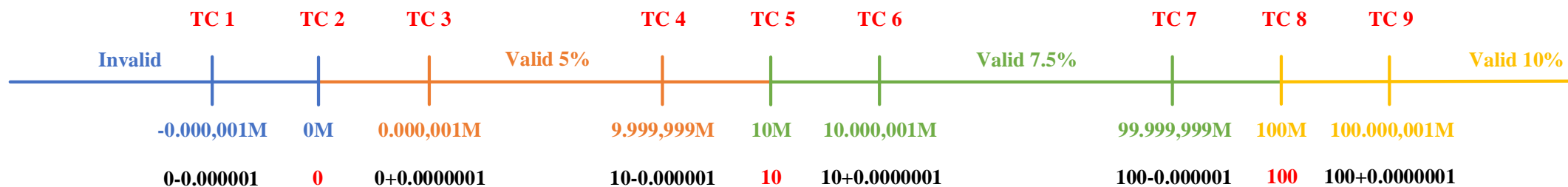


Ví dụ minh họa





Ví dụ minh họa



Cách 1:

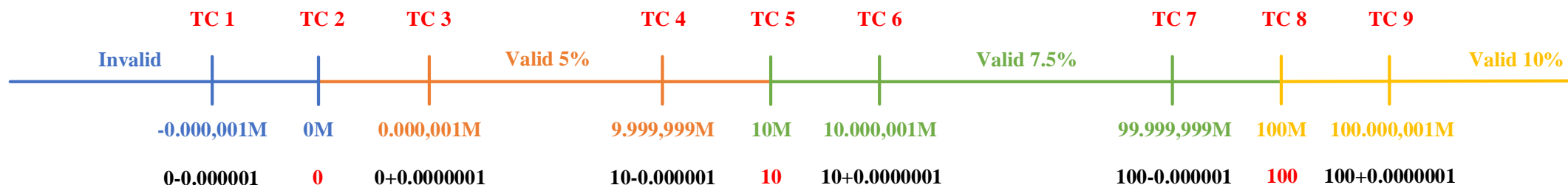
$n = \{0; 10; 100\} = 3$

$$Total\ TCs = 3 * n = 3 * 3 = 9$$





Ví dụ minh họa



Cách 2:

$$\begin{aligned} \text{Total TCS} &= \sum_{i=1}^3 m_{1(0)} + m_{2(10)} + m_{3(100)} \\ &= 3 + 3 + 3 = 9 \end{aligned}$$





06

Ưu nhược điểm của BVA





Ưu nhược điểm

#	Ưu điểm (Advantages)	Nhược điểm (Disadvantages)
1	Tập trung vào việc kiểm tra các giá trị biên quan trọng nhất của phạm vi đầu vào, nơi lỗi thường xuất hiện.	BVA chỉ phù hợp khi phạm vi đầu vào có giới hạn rõ ràng và các biên được xác định một cách chính xác.
2	Tiết kiệm thời gian và nguồn lực vì BVA chỉ kiểm tra các giá trị cận biên quan trọng nhất.	Dù BVA giúp tập trung vào các giá trị quan trọng, nhưng nó không đảm bảo phát hiện được tất cả các loại lỗi trong phần mềm, đồng thời gây ra rủi ro cho chất lượng phần mềm.
3	Triển khai một cách dễ dàng và thực hiện một cách hiệu quả, đặc biệt là trong các trường hợp phạm vi đầu vào có số lượng lớn giá trị.	





B A T I Z E N S



Trân trọng cảm ơn!

