

Bài tập Testing

Hoàng Ngọc Quân

October 2023

1 Bài 1

Problem 1. Cho đoạn code sau đây, hãy viết các bộ test sử dụng phương pháp *Condition Coverage* để kiểm tra toàn bộ các điều kiện, trường hợp của đoạn code sau:

```
def is_triangle(a, b, c):  
    if a <= 0 (1) or b <= 0 (2) or c <= 0 (3):  
        return "Not a triangle"  
    elif a + b <= c (4) or a + c <= b (5) or b + c <= a (6):  
        return "Not a triangle"  
    elif a == b (7) and b == c (8):  
        return "Equilateral triangle"  
    elif a == b (9) or b == c (10) or c == a (11):  
        return "Isosceles triangle"  
    else:  
        return "Scalene triangle"
```

Có tất cả 11 điều kiện cần bao phủ được đánh số như bên trên.

Solution. Ta có bảng testcase kiểm tra một cách tuần tự tuyến tính các điều kiện như sau:

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
(0, 0, 0)	T	X	X	X	X	X	X	X	X	X	X
(1, 0, 0)	F	T	X	X	X	X	X	X	X	X	X
(1, 1, 0)	F	F	T	X	X	X	X	X	X	X	X
(1, 1, 2)	F	F	F	T	X	X	X	X	X	X	X
(1, 2, 1)	F	F	F	F	T	X	X	X	X	X	X
(2, 1, 1)	F	F	F	F	F	T	X	X	X	X	X
(1, 1, 1)	F	F	F	F	F	F	T	T	X	X	X
(2, 2, 1)	F	F	F	F	F	F	T	F	T	X	X
(1, 2, 2)	F	F	F	F	F	F	F	X	F	T	X
(2, 1, 2)	F	F	F	F	F	F	F	X	F	F	T
(2, 3, 4)	F	F	F	F	F	F	F	X	F	F	F

Trong đó: T: trường hợp đúng, F: trường hợp sai, X: không cần xét đến

□

output của từng testcase tương ứng:

- (1): "Not a triangle"
- (2): "Not a triangle"
- (3): "Not a triangle"
- (4): "Not a triangle"
- (5): "Not a triangle"
- (6): "Not a triangle"
- (7): "Equilateral triangle"
- (8): "Isosceles triangle"

- (9): "Isosceles triangle"
- (10): "Isosceles triangle"
- (11): "Scalene triangle"

2 Bài 2

Vì đề không ghi rõ các giới hạn, vì vậy chỉ có thể mô tả các testcase mà không có testcase cụ thể.

2.1 Câu a

Testcase 1: Không có số chẵn nào

- input: toàn bộ đều là số lẻ (số -1)
- output: 0

Testcase 2: Tất cả đều là số chẵn, nhưng có số 0

- input: toàn bộ đều là số 0 và có thêm một số chẵn $2k$
- output: $2k$

Testcase 3: Tất cả đều là số chẵn, nhưng không có số 0

- input: toàn bộ đều là số chẵn (số -2)
- output: $-2N$

Testcase 4: Chỉ có một số chẵn duy nhất

- input: toàn bộ đều là số lẻ (số -1) và có thêm một số $-2k$
- output: $-2k$

Testcase 5: Có $N - 1$ số chẵn

- input: toàn bộ đều là số chẵn (số -2) nhưng có thêm một số 1
- output: $-2(N - 1)$

Vì bộ testcase trên phủ hết tất cả các trường hợp biên theo tiêu chí Boundary Value Testing. Đó là các trường hợp biên mà chỉ có 0, 1, $N-1$, N số chẵn và các trường hợp dữ liệu đặc biệt như: chứa số âm, số 0.

2.2 Câu b

Ta chia dữ liệu vào thành các lớp tương ứng với các trường hợp:

Gọi N là độ dài của chuỗi kí tự.

- N chẵn: thỏa mãn và không thỏa mãn.
- N lẻ: thỏa mãn và không thỏa mãn.
- N đặc biệt (0, 1) và các kí tự đặc biệt (' ', '@')

Testcase 1:

- input: N chẵn, chỉ gồm các kí tự 'a' (aaaaaa)
- output: thỏa mãn

Testcase 2:

- input: N chẵn, gồm k kí tự 'a', 1 kí tự 'b', $k-1$ kí tự 'a' (aaabaa)
- output: không thỏa mãn

Testcase 3:

- input: N lẻ, gồm k kí tự 'a', 1 kí tự 'b', k kí tự 'a' (aaabaaa)
- output: thỏa mãn

Testcase 4:

- input: N lẻ, gồm $N-1$ kí tự 'a', 1 kí tự 'b' (aaaaaab)
- output: không thỏa mãn

Testcase 5:

- input: chỉ gồm một chuỗi rỗng ("")
- output: thỏa mãn

Testcase 6:

- input: chỉ gồm một kí tự '@' (@)

- output: thỏa mãn

Testcase 7:

- input: chỉ gồm 2k-1 ký tự khoảng trắng ' ' và 1 ký tự '@' (@)

- output: không thỏa mãn

Bộ testcase trên chia chuỗi đầu vào thành các lớp (trường hợp) và lấy đại diện của từng trường hợp để kiểm tra theo tiêu chí Equivalence Class Testing: bao gồm độ dài chẵn, lẻ, thỏa mãn và không thỏa mãn. Ngoài ra còn có thêm các trường hợp đặc biệt mà chuỗi có độ dài hoặc chứa ký tự đặc biệt làm ảnh hưởng đến việc nhập xuất cũng rất nên được kiểm tra.