**BTVN Kiểm thử dòng điều khiển**

**Câu 1:** Trình bày các bước nhằm kiểm thử một đơn vị chương trình theo phương pháp kiểm thử dòng điều khiển với một độ đo kiểm thử cho trước.

Quy trình kiểm thử một đơn vị chương trình theo phương pháp dòng điều khiển với một độ đo kiểm thử cho trước là:

**Bước 1. Xây dựng đồ thị dòng điều khiển (Control Flow Graph – CFG)**

- Từ đơn vị chương trình cần kiểm thử và độ đo kiểm thử đã được chọn (ví dụ: C1 – phủ câu lệnh, C2 – phủ nhánh, C3 – phủ điều kiện), tiến hành xây dựng đồ thị dòng điều khiển tương ứng.

- Trong đồ thị:

+ Các đỉnh (nodes) biểu diễn các lệnh hoặc khối lệnh tuần tự.

+ Các cạnh (edges) biểu diễn luồng điều khiển giữa các khối lệnh.

- Lưu ý rằng: CFG cho các độ đo C1 và C2 thường giống nhau, trong khi CFG cho độ đo C3 có thể khác do cần xem xét chi tiết các điều kiện con trong biểu thức logic.

**Bước 2. Xác định các đường đi kiểm thử (Test Paths)**

- Dựa trên đồ thị dòng điều khiển đã xây dựng, xác định tập các đường đi (paths) sao cho thỏa mãn tiêu chí của độ đo kiểm thử đã chọn.

- Ví dụ:

+ Với C1 (Statement Coverage): các đường đi phải bao phủ toàn bộ các câu lệnh trong chương trình.

+ Với C2 (Branch Coverage): các đường đi phải bao phủ toàn bộ các nhánh (true/false) của các điều kiện rẽ nhánh.

+ Với C3 (Condition Coverage): các đường đi phải làm cho mỗi điều kiện con nhận cả hai giá trị đúng và sai ít nhất một lần.

**Bước 3. Sinh ca kiểm thử (Generate Test Cases)**

- Mỗi đường đi độc lập được xác định ở bước trước sẽ tương ứng với một ca kiểm thử (test case).

- Quá trình sinh ca kiểm thử bao gồm hai hoạt động chính:

1. Xác định dữ liệu đầu vào:

+ Lựa chọn các giá trị đầu vào cụ thể sao cho, khi chương trình thực thi, luồng điều khiển sẽ đi đúng theo đường đi đã chọn.

+Việc này đòi hỏi phải làm thỏa mãn các điều kiện tại những điểm quyết định (các nút rẽ nhánh) trên đường đi.

2. Xác định kết quả mong đợi:

+ Với mỗi bộ dữ liệu đầu vào, người kiểm thử cần dự đoán và ghi lại kết quả đầu ra mong đợi của chương trình.

+ Kết quả này được dùng để đối chiếu với kết quả thực tế trong quá trình kiểm thử.

**Bước 4. Thực hiện các ca kiểm thử (Execute Test Cases)**

- Tiến hành chạy chương trình với các bộ dữ liệu đầu vào đã thiết kế ở bước trước.

- Trong quá trình thực thi:

+ Ghi nhận đường đi thực tế mà chương trình đã thực hiện.

+ So sánh kết quả đầu ra thực tế với kết quả mong đợi để phát hiện sai sót hoặc điểm bất thường trong hành vi của chương trình.

- Nếu kết quả thực tế khác với kết quả mong đợi, lỗi trong mã nguồn sẽ được ghi nhận để phân tích và khắc phục.

**Câu 2:**

A diagram of a graph

AI-generated content may be incorrect.

Đường đi và ca kiểm thử với độ đo C1 (Phủ câu lệnh):

Path 1: 1 → 2(T) → 3

Test case 1: Grade(11)

Path 2: 1 → 2(F) → 4(T) → 5 → 13

Test case 2: Grade(9.5)

Path 3: 1 → 2(F) → 4(F) → 6(T) → 7 → 13

Test case 3: Grade(8.5)

Path 4: 1 → 2(F) → 4(F) → 6(F) → 8(T) → 9 → 13

Test case 4: Grade(7)

Path 5: 1 → 2(F) → 4(F) → 6(F) → 8(F) → 10(T) → 11 → 13

Test case 5: Grade(6)

Path 6: 1 → 2(F) → 4(F) → 6(F) → 8(F) → 10(F) → 12 → 13

Test case 6: Grade(4)

Đường đi và ca kiểm thử với độ đo C2 (Phủ nhánh)

Path 1: 1 → 2(T) → 3

Test case 1: Grade(11)

Path 2: 1 → 2(F) → 4(T) → 5 → 13

Test case 2: Grade(9.5)

Path 3: 1 → 2(F) → 4(F) → 6(T) → 7 → 13

Test case 3: Grade(8.5)

Path 4: 1 → 2(F) → 4(F) → 6(F) → 8(T) → 9 → 13

Test case 4: Grade(7)

Path 5: 1 → 2(F) → 4(F) → 6(F) → 8(F) → 10(T) → 11 → 13

Test case 5: Grade(6)

Path 6: 1 → 2(F) → 4(F) → 6(F) → 8(F) → 10(F) → 12 → 13

Test case 6: Grade(4)

**Câu 3:**

A diagram of a function

AI-generated content may be incorrect.

Đường đi và ca kiểm thử với độ đo C1 (Phủ câu lệnh):

Path 1: 1 → 2(F) → 4

Test case 1: Sum(a, 0) với a = {} và n = 0

Path 2: 1 → 2(T) → 3 → 2(F) → 4

Test case 2: Sum(a, 1) với a = {10} và n = 1

Đường đi và ca kiểm thử với độ đo C2 (Phủ nhánh)

Path 1: 1 → 2(F) → 4

Test case 1: Sum(a, 0)

Path 2: 1 → 2(T) → 3 → 2(F) → 4

Test case 2: Sum(a, 1)

Các ca kiểm thử cho vòng lặp for:

- Vòng lặp không chạy lần nào (0 lần):

+ Ca kiểm thử: Sum(a, 0) với a = {}, n = 0.

+ Kết quả mong đợi: 0

- Vòng lặp chạy đúng 1 lần:

+ Ca kiểm thử: Sum(a, 1) với a = {5}, n = 1.

+ Kết quả mong đợi: 5

- Vòng lặp chạy 2 lần:

+ Ca kiểm thử: Sum(a, 2) với a = {5, 10}, n = 2.

+ Kết quả mong đợi: 15

- Vòng lặp chạy k lần (một số lần tiêu biểu, 1 < k < max):

+ Ca kiểm thử: Sum(a, 3) với a = {1, 2, 3}, n = 3.

+ Kết quả mong đợi: 6

**Câu 4:**

A diagram of a algorithm

AI-generated content may be incorrect.

Đường đi và ca kiểm thử với độ đo C2 (Phủ nhánh):

Path 1: 1 -> 2 -> 3 -> 7

Test case 1: foo(65)

Path 2: 1 -> 2 -> 4 -> 7

Test case 2: foo(66)

Path 3: 1 -> 2 -> 5 -> 7

Test case 3: foo(67)

Path 4: 1 -> 2 -> 6 -> 7

Test case 4: foo(68)

**Câu 5:** Kiểm thử chương trình của bạn với độ đo kiểm thử C2

Bài toán tự đặt ra: Một rạp chiếu phim áp dụng chính sách giá vé như sau: Giá vé gốc là 100,000 VND. Khách hàng được giảm giá 20% nếu dưới 12 tuổi hoặc từ 60 tuổi trở lên. Khách hàng được giảm giá 10% nếu là sinh viên. Trong trường hợp một khách hàng vừa là sinh viên, vừa thuộc nhóm tuổi được giảm, thì chỉ áp dụng mức giảm cao nhất, không cộng dồn.

Chương trình nhận vào hai biến: age (số nguyên, biểu thị tuổi của khách hàng) và isStudent (kiểu boolean, cho biết khách hàng có phải sinh viên hay không). Kết quả đầu ra là số tiền phải trả.

A diagram of a program

AI-generated content may be incorrect.

Đường đi và ca kiểm thử với độ đo C2 (Phủ nhánh):

Path 1: (1,2,3) → 4(T) → 5

Test case 1: tuoi = -5

Path 2: (1,2,3) → 4(F) → (6,7) → 8(catch) → 9

Test case 2: tuoi = 20, isStudent = "abc"

Path 3: (1,2,3) → 4(F) → (6,7) → 8 → 10(T) → 11 → 12(T) → 13 → 14

Test case 3: tuoi = 10, isStudent = true

Path 4: (1,2,3) → 4(F) → (6,7) → 8 → 10(T) → 11 → 12(F) → 14

Test case 4: tuoi = 25, isStudent = true

Path 5: (1,2,3) → 4(F) → (6,7) → 8 → 10(F) → 12(F) → 14

Test case 5: tuoi = 30, isStudent = false

Path 6: (1,2,3) → 4(F) → (6,7) → 8 → 10(F) → 12(T) → 13 → 14

Test case 6: tuoi = 65, isStudent = false