

KIỂM THỬ DÒNG ĐIỀU KHIỂN

INT3117 40

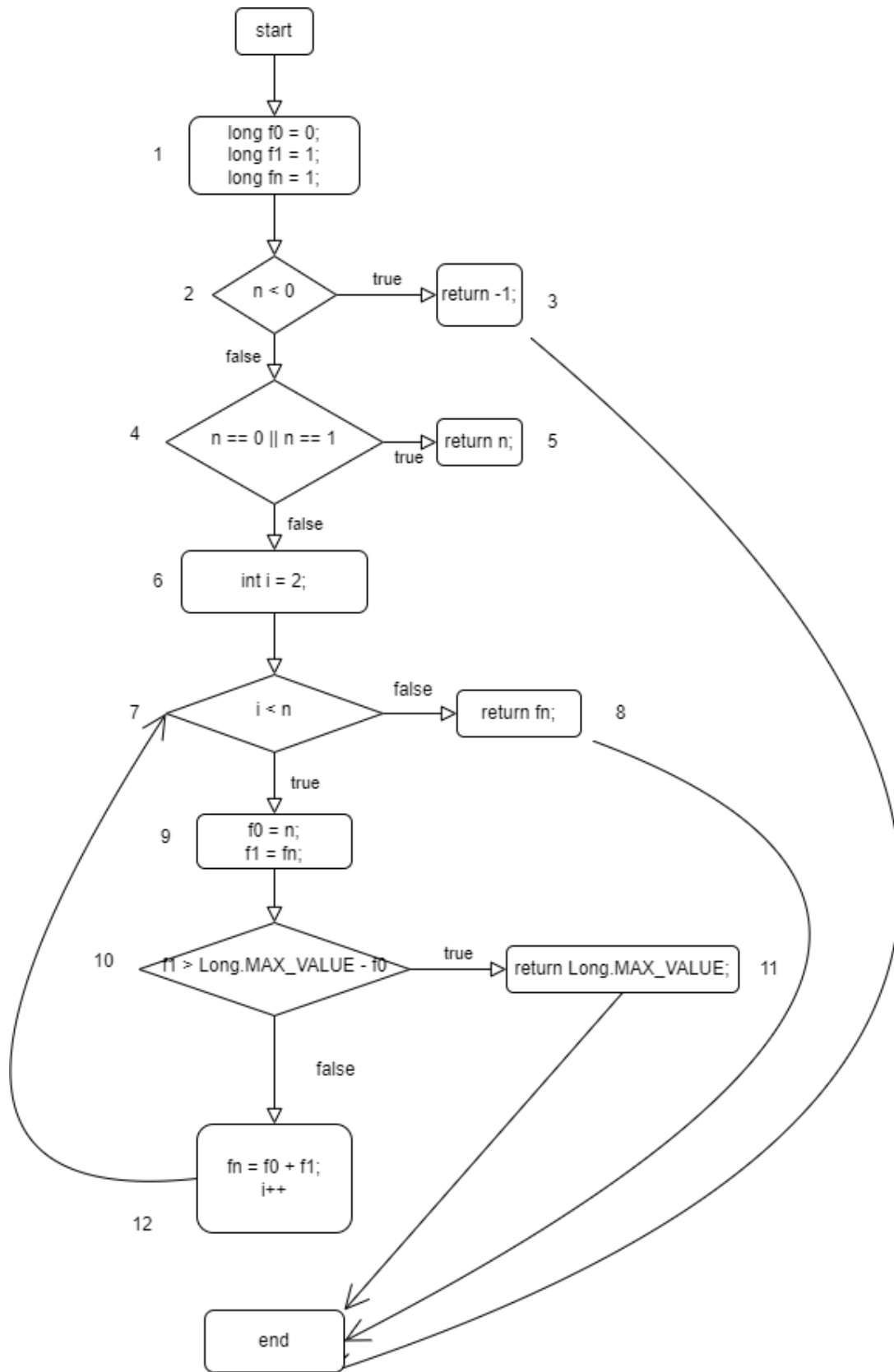
Họ và tên: Phạm Văn Tiến

Mssv: 20021449

1. Source code: [KiemThu/Code.java at main · Tienpv-02/KiemThu \(github.com\)](#)

```
public static long fibonacci(long n) {  
    long f0 = 0;  
    long f1 = 1;  
    long fn = 1;  
    if (n < 0) {  
        return -1;  
    } else if (n == 0 || n == 1) {  
        return n;  
    } else {  
        for (int i = 2; i < n; i++) {  
            f0 = f1;  
            f1 = fn;  
            if (f1 > Long.MAX_VALUE - f0) {  
                return Long.MAX_VALUE;  
            }  
            fn = f0 + f1;  
        }  
    }  
    return fn;  
}
```

2. Đồ thị dòng điều khiển (CFG)



3. Sinh các đường đi với độ đo C2

- 1) start → 1 → 2(T) → 3 → end
- 2) start → 1 → 2(F) → 4(T) → 5 → end
- 3) start → 1 → 2(F) → 4(F) → 6 → 7(F) → 8 → end
- 4) start → 1 → 2(F) → 4(T) → 6 → 7(T) → 9 → 10(T) → 11 → end
- 5) start → 1 → 2(F) → 4(T) → 6 → 7(T) → 9 → 10(F) → 12 → 7(F) → 8 → end

4. Sinh các ca kiểm thử

- 1) n = -1
- 2) n = 0
- 3) n = 2
- 4) n = 100
- 5) n = 3

5. Thực hiện các ca kiểm thử

```
Pham Van Tien *
@Test
public void testFibonacci_1() {
    assertEquals( expected: -1, Code.fibonacci( n: -1));
}

Pham Van Tien
@Test
public void testFibonacci_2() {
    assertEquals( expected: 0, Code.fibonacci( n: 0));
}

Pham Van Tien *
@Test
public void testFibonacci_3() {
    assertEquals( expected: 1, Code.fibonacci( n: 2));
}

Pham Van Tien *
@Test
public void testFibonacci_4() {
    assertEquals(Long.MAX_VALUE, Code.fibonacci( n: 100));
}

Pham Van Tien *
@Test
public void testFibonacci_5() {
    assertEquals( expected: 2, Code.fibonacci( n: 3));
}
}
```

Kết quả chạy

✓ CodeTest	2 ms
✓ testFibonacci_1	2 ms
✓ testFibonacci_2	0 ms
✓ testFibonacci_3	0 ms
✓ testFibonacci_4	0 ms
✓ testFibonacci_5	0 ms

6. Bảng kiểm thử

Test case	Test Path	Input	Expected	Actual	Result
1	start → 1 → 2(T) → 3 → end	-1	-1	-1	pass
2	start → 1 → 2(F) → 4(T) → 5 → end	0	0	0	pass
3	start → 1 → 2(F) → 4(F) → 6 → 7(F) → 8 → end	2	1	1	pass
4	start → 1 → 2(F) → 4(T) → 6 → 7(T) → 9 → 10(T) → 11 → end	100	Long.MAX VALUE	Long.MAX VALUE	pass
5	start → 1 → 2(F) → 4(T) → 6 → 7(T) → 9 → 10(F) → 12 → 7(F) → 8 → end	3	2	2	pass