

CN 202 – Data Structures and Algorithms I

Solution for Homework 11

1. (ข้อสอบปลายภาคปี 47)

กำหนดโปรแกรมในภาษา Java ให้ดังต่อไปนี้

```
public class FunTest
{
    public static void main(String[] args)
    {
        int i = Integer.parseInt(args[0]);
        System.out.println("fun(" + i + ")");
        fun(i);
        System.out.println();
    }
    public static void fun(int n)
    {
        if (n > 0)
            if (n == 1)
                System.out.print("X");
            else if (n == 2)
                System.out.print("Y");
            else
            {
                System.out.print("A");
                fun(n-1);
                System.out.print("B");
                fun(n-2);
                System.out.print("C");
            }
    }
}
```

หา Output ของโปรแกรม ถ้าขณะที่ทำการรันโปรแกรม เรามีการป้อนค่าที่โปรแกรมต้องการ ผ่านทาง Command Line ดังนี้

(ก) % java FunTest 4

Output ของโปรแกรม คือ

fun(4)

AAYBXCBYC

(ข) % java FunTest 5
 Output ของโปรแกรม คือ
 fun(5)
 AAAYBXCBYCBAYBXCC

6. (ข้อสอบปลายภาคปี 52) กำหนด Static Method ที่ชื่อว่า T ให้ดังต่อไปนี้

```
public static int T(int n)
{
    if (n == 1)
        return 0;
    else
        return 2*T(n-1) + 2;
}
```

(ข) หา $T(n)$ ในเทอมของ n

จาก Static Method ที่กำหนดให้ เราสามารถเขียน $T(n)$ ในรูปของ Recurrence Relation ได้ดังนี้

$$T(n) = 2 * T(n-1) + 2 \quad \text{เมื่อ } n > 1$$

$$T(n) = 0 \quad \text{เมื่อ } n = 1$$

$$T(1) = 0$$

$$T(2) = 2 * T(1) + 2 = 2 * 0 + 2 = 2$$

$$T(3) = 2 * T(2) + 2 = 2 * 2 + 2 = 6$$

$$T(4) = 2 * T(3) + 2 = 2 * 6 + 2 = 14$$

$$T(5) = 2 * T(4) + 2 = 2 * 14 + 2 = 30$$

...

$$T(n) = 2^n - 2$$

(ก) หา Output ของ Method ถ้าค่า $n = 10$ ถูกส่งให้เป็น Input ของ Method

จากข้อ (ข) เมื่อ $n = 10$

$$T(10) = 2^{10} - 2 = 1022$$

8. (ข้อสอบปลายภาคปี 2563) กำหนดโปรแกรมในภาษาจาวาให้ดังต่อไปนี้

```
public class TestRecursive
{
    public static int fun(int n)
    {
        int x = 1;
        if (n == 1)
            return x;
        for (int k = 1; k < n; k++)
            x += fun(k) * fun(n-k);
        return x;
    }

    public static void main(String[] args)
    {
        System.out.println(fun(5));
    }
}
```

จากโปรแกรมที่ให้มา จะได้ว่า

$\text{fun}(1) = 1$

การเรียก $\text{fun}(1)$ 1 ครั้ง Method fun จะถูกเรียกใช้งานทั้งหมด 1 ครั้ง

ในการหาค่า $\text{fun}(2)$

เมื่อเข้าสู่การทำซ้ำรอบที่ $k = 1$ จะได้

$$x = x + (\text{fun}(1) * \text{fun}(2-1)) = 1 + (\text{fun}(1) * \text{fun}(1)) = 1 + (1 * 1) = 2$$

ซึ่งจะได้ $\text{fun}(2) = 2$

โดยจะเห็นว่า $\text{fun}(2)$ เรียก $\text{fun}(1)$ จำนวน 2 ครั้ง กล่าวคือ การเรียก $\text{fun}(2)$ 1 ครั้ง Method fun จะถูกเรียกใช้งานทั้งหมด = จำนวนครั้งที่เรียก $\text{fun}(2)$ + จำนวนครั้งที่เรียก $\text{fun}(1) = 1 + (2 * 1) = 3$ ครั้ง

ในการหาค่า $\text{fun}(3)$

เมื่อเข้าสู่การทำซ้ำรอบที่ $k = 1$ จะได้

$$x = x + (\text{fun}(1) * \text{fun}(3-1)) = x + (\text{fun}(1) * \text{fun}(2)) = 1 + (1 * 2) = 3$$

เมื่อเข้าสู่การทำซ้ำรอบที่ $k = 2$ จะได้

$$x = x + (\text{fun}(2) * \text{fun}(3-2)) = x + (\text{fun}(2) * \text{fun}(1)) = 3 + (2 * 1) = 5$$

ซึ่งจะได้ $\text{fun}(3) = 5$

โดยจะเห็นว่า $\text{fun}(3)$ เรียก $\text{fun}(2)$ จำนวน 2 ครั้ง และเรียก $\text{fun}(1)$ จำนวน 2 ครั้ง กล่าวคือ การเรียก $\text{fun}(3)$ 1 ครั้ง Method fun จะถูกเรียกใช้งานทั้งหมด = จำนวนครั้งที่ $\text{fun}(3)$ ถูกเรียก + จำนวนครั้งที่ $\text{fun}(2)$ ถูกเรียก + จำนวนครั้งที่ $\text{fun}(1)$ ถูกเรียก = $1 + (2 * 3) + (2 * 1) = 9$ ครั้ง

ในการหาค่า fun(4)

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 1 จะได้

$$x = x + (\text{fun}(1) * \text{fun}(4-1)) = x + (\text{fun}(1) * \text{fun}(3)) = 1 + (1 * 5) = 6$$

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 2 จะได้

$$x = x + (\text{fun}(2) * \text{fun}(4-2)) = x + (\text{fun}(2) * \text{fun}(2)) = 6 + (2 * 2) = 10$$

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 3 จะได้

$$x = x + (\text{fun}(3) * \text{fun}(4-3)) = x + (\text{fun}(3) * \text{fun}(1)) = 10 + (5 * 1) = 15$$

ซึ่งจะได้ fun(4) = 15

โดยจะเห็นว่า fun(4) เรียก fun(3) จำนวน 2 ครั้ง เรียก fun(2) จำนวน 2 ครั้ง และเรียก fun(1) จำนวน 2 ครั้ง กล่าวคือ การเรียก fun(4) 1 ครั้ง Method fun จะถูกเรียกใช้งานทั้งหมด = จำนวนครั้งที่ fun(4) ถูกเรียก + จำนวนครั้งที่ fun(3) ถูกเรียก + จำนวนครั้งที่ fun(2) ถูกเรียก + จำนวนครั้งที่ fun(1) ถูกเรียก = $1 + (2 * 9) + (2 * 3) + (2 * 1) = 1 + 18 + 6 + 2 = 27$ ครั้ง

ในการหาค่า fun(5)

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 1 จะได้

$$x = x + (\text{fun}(1) * \text{fun}(5-1)) = x + (\text{fun}(1) * \text{fun}(4)) = 1 + (1 * 15) = 16$$

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 2 จะได้

$$x = x + (\text{fun}(2) * \text{fun}(5-2)) = x + (\text{fun}(2) * \text{fun}(3)) = 16 + (2 * 5) = 26$$

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 3 จะได้

$$x = x + (\text{fun}(3) * \text{fun}(5-3)) = x + (\text{fun}(3) * \text{fun}(2)) = 26 + (5 * 2) = 36$$

เมื่อเข้าสู่การทำซ้ำรอบที่ k = 4 จะได้

$$x = x + (\text{fun}(4) * \text{fun}(5-4)) = x + (\text{fun}(4) * \text{fun}(1)) = 36 + (15 * 1) = 51$$

ซึ่งจะได้ fun(5) = 51

โดยจะเห็นว่า fun(5) เรียก fun(1) จำนวน 2 ครั้ง เรียก fun(2) จำนวน 2 ครั้ง เรียก fun(3) จำนวน 2 ครั้ง และเรียก fun(4) จำนวน 2 ครั้ง กล่าวคือ การเรียก fun(5) 1 ครั้ง Method fun จะถูกเรียกใช้งานทั้งหมด = จำนวนครั้งที่ fun(5) ถูกเรียก + จำนวนครั้งที่ fun(4) ถูกเรียก + จำนวนครั้งที่ fun(3) ถูกเรียก + จำนวนครั้งที่ fun(2) ถูกเรียก + จำนวนครั้งที่ fun(1) ถูกเรียก = $1 + (2 * 27) + (2 * 9) + (2 * 3) + (2 * 1) = 1 + 54 + 18 + 6 + 2 = 81$ ครั้ง

(ก) แสดงการหา Output ของโปรแกรม

Output ของโปรแกรม คือ ค่าของ fun(5) ซึ่งเท่ากับ 51

(ข) จากการทำงานของโปรแกรมที่ให้มา แสดงการหาให้ดูว่า Method fun ถูกเรียกใช้งานรวมทั้งหมดกี่ครั้ง

การทำงานของโปรแกรมจะมีการเรียก fun(5) 1 ครั้ง ซึ่งจะมี Method fun ถูกเรียกใช้ทั้งหมด 81 ครั้ง