Programming competition guidebook

Juan J. Alvarez
juan.alvarez7@upr.edu
UPR Bayamón
Computer Science Department

Contents

1 Mathematics **2**
1.1 Check whether a number is prime . . . 2
1.1.1 Pseudocode . . . 2
1.1.2 Implementation . . . 2
1.2 List the divisors of a number . . . 2
1.2.1 Pseudocode . . . 2
1.2.2 Implementation . . . 3
1.3 Calculating Factorials . . . 3
1.3.1 Pseudocode . . . 3
1.3.2 Implementation . . . 3
1.3.3 BigInteger Implementation . . . 3
1.4 Sum of Natural Numbers . . . 4
1.5 Divisibility Rules . . . 4

1 Mathematics

1.1 Check whether a number is prime

In this section we talk about how to programatically check the primality of a given number.

1.1.1 Pseudocode

Algorithm 1 Prime check algorithm

```
1: procedure isPrime(n)
2:     if n < 2 then return false
3:     if n == 2 then return true
4:     if n%2 == 0 then return false
5:     for x = 3 ; x^2 <= n ; x+ = 2 do
6:         if n%x == 0 then return false
       return true
```

1.1.2 Implementation

```
public static boolean isPrime(int n) {
        if (n < 2)
                return false;
        if (n == 2)
                return true;
        if (n % 2 == 0)
                return false;
        for (int x = 3; x * x <= n; x += 2)
                if (n % x == 0)
                        return false;
        return true;
}
```

1.2 List the divisors of a number

In this section we talk about how to programatically list the divisors of a given number.

1.2.1 Pseudocode

Algorithm 2 Divisor list algorithm

```
1: procedure listDivisors(n)
2:     list = new empty list of numbers
3:     add 1 to list
4:     mpd = √n
5:     for x = 2 ; x <= mpd ; x + + do
6:         if n%x == 0 then
7:             add x to list
8:             if n ÷ x ≠ x then
9:                 add n/d to list
10:    add n to list
11: return list
```

1.2.2 Implementation

```
private static ArrayList<Integer> getDivisors(int input) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        list.add(1);
        int maxD = (int) Math.sqrt(input);
        for (int i = 2; i <= maxD; i++) {
                if (input % i == 0) {
                        list.add(i);
                        int d = input / i;
                        if (d != i)
                                list.add(d);
                }
        }
        list.add(input);
        return list;
}
```

1.3 Calculating Factorials

1.3.1 Pseudocode

Algorithm 3 Factorial Algorithm

```
1: procedure FACTORIAL(n)
2:     total = 1
3:     for x = n ; x > 1 ; x++ do
4:         total *= x
       return total
```

1.3.2 Implementation

This method has a huge flaw due to the fact that it uses 64-bit integers to calculate the result, it is not capable of calculating the factorial of any number larger than 16.

```
public static long fac(int n) {
        long total = 1;
        for (int x = n; x > 1; x--)
                total *= n;
        return total;
}
```

1.3.3 BigInteger Implementation

This implementation was done using the BigInteger class to solve the limitational problems of using 64-bit integers.

```
public static BigInteger fac(BigInteger n) {
        BigInteger total = BigInteger.ONE, x;
        for (x = n; x.compareTo(BigInteger.ONE) == 1; x = x.subtract(BigInteger.ONE))
                total = total.multiply(n);
        return total;
}
```

1.4 Sum of Natural Numbers

The sum of the sequence of natural numbers $\{1, 2, 3, ..., n\}$ can be written as

$$\sum_{x=1}^{n} x = \frac{n(n+1)}{2}$$

.

1.5 Divisibility Rules

2	The last digit is 0, 2, 4, 6 or 8. Or the modulus operation with 2 yields 0.
3	The sum of the digits is divisible by 3.
4	The number formed by the last two digits is divisible by 4.
5	The last digit is either 0 or 5.
6	It is divisible by 2 AND it is divisible by 3.
7	If the last digit multiplied by two and subtracted from the rest of the number is divisible by 7.
8	The last three digits are divisible by 8.
9	The sum of the digits is divisible by 9.
10	The last digit is 0.
11	The difference between the sum of the odd placed digits and the sum of the even placed digits is divisible by 11.
12	The number is divisible by both 3 and 4.
13	Subtract 9 times the last digit from the rest of the number, the result is divisible by 13.
14	It is divisible by 2 and 7.
15	It is divisible by 3 and 5.
16	The last 4 digits are divisible by 16.