Teoría de Números

Sofía Martin

Facultad de Informática

TTPS 2016

Contenido

Aritmética Modular

Módulo: esta operación devuelve el resto de la división, cuando un número es dividido por otro. Se utiliza el signo %.

Ejemplo

Si tenemos dos números 5 y 2, entonces 5%2 es 1, ya que el resto de la división es 1.

Propiedades

$(a + b)\%c = (a\%c + b\%c)\%c$. $(a * b)\%c = ((a\%c) * (b\%c))\%c$.

Ejemplo:

a = 5, b = 3, c = 2.
$(5 + 3)\%2 = 8\%2 = 0$.
$(5\%2 + 3\%2)\%2 = (1 + 1)\%2 = 0$.
$(5 * 3)\%2 = 15\%2 = 1$.
$((5\%2) * (3\%2))\%2 = (1 * 1)\%2 = 1$.

Exponenciación Modular - Uso

- Exponenciación modular son considerados **fáciles** de resolver.
- Exponenciación modular **difícil** de encontrar b si es dado un a, c, y $res \Longrightarrow$ **Función unidireccional**.
- **Función unidireccional** candidato para su uso en algoritmos criptográficos.

Exponenciación Modular

Supongamos tenemos que calcular $a^b\%c$, donde % es el módulo y b puede ser un número muy grande. (por ej. alrededor de 1018).

```
long long exponenciacion(long long a, long long b, long long c) {
        long long res = 1;
        for(int i = 1;i <= b;i++) {
            res *= a;                          //multiplicando a, b veces.
            res %= c;
        }
    return res;
}
```

El tiempo de ejecución es de $\mathcal{O}(b)$

Analicemos una técnica denominada **exponenciación binaria o potenciación por cuadrados** que usa $\mathcal{O}(\log_2(b))$ multiplicaciones.

Algunas propiedades de la potencia que nos pueden ayudar :

- $x^1 = x$
- $x^{a+b} = x^a x^b$
- $x^{ab} = (x^a)^b$

Exponenciación Modular

Supongamos tenemos que calcular $a^b \% c$, donde % es el módulo y b puede ser un número muy grande. (por ej. alrededor de 1018).

```
long long exponenciacion(long long a, long long b, long long c) {
        long long res = 1;
        for(int i = 1;i <= b;i++) {
            res *= a;                          //multiplicando a, b veces.
            res %= c;
        }
    return res;
}
```

El tiempo de ejecución es de $\mathcal{O}(b)$
Analicemos una técnica denominada **exponenciación binaria o potenciación por cuadrados** que usa $\mathcal{O}(\log_2(b))$ multiplicaciones.
Algunas propiedades de la potencia que nos pueden ayudar :

- $x^1 = x$
- $x^{a+b} = x^a x^b$
- $x^{ab} = (x^a)^b$

Exponenciación Modular

Supongamos tenemos que calcular $a^b\%c$, donde % es el módulo y b puede ser un número muy grande. (por ej. alrededor de 1018).

```
long long exponenciacion(long long a, long long b, long long c) {
        long long res = 1;
        for(int i = 1;i <= b;i++) {
            res *= a;                         //multiplicando a, b veces.
            res %= c;
        }
    return res;
}
```

El tiempo de ejecución es de $\mathcal{O}(b)$
Analicemos una técnica denominada **exponenciación binaria o potenciación por cuadrados** que usa $\mathcal{O}(\log_2(b))$ multiplicaciones.
Algunas propiedades de la potencia que nos pueden ayudar :

- $x^1 = x$
- $x^{a+b} = x^a x^b$
- $x^{ab} = (x^a)^b$

Entendiendo exponenciación modular

Veamos cómo se descompone b para poder realizar a^b. Tomemos $b = 45$, lo expresamos en binario: $b = 101101_2$, prestar atención a los **1**, nos queda $a^b = a^{101101_2}$

Utilizando la representación binaria de b,

$$b = 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 .$$

Los 0 anulan la posición correspondiente en la representación binaria, por lo tanto no se tienen en cuenta. Nos queda:

$$a^b = a^{1*2^5} * a^{0*2^4} * a^{1*2^3} * a^{1*2^2} * a^{0*2^1} * a^{1*2^0}$$
$$a^b = a^{2^5} * 1 * a^{2^3} * a^{2^2} * 1 * a^{2^0}$$
$$a^b = a^{32} * a^8 * a^4 * a$$

Podemos realizar el cálculo en 5 iteraciones. En cada iteración el valor de 'a' se convierte en $a * a$.

Ejemplo exponenciación modular

Calcular $5^{59}\%19$. En primer lugar representamos en binario a 59.

$5^{59} = 5^{111011_2} = 5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0}$

Llegaremos al resultado en 6 iteraciones:

Definamos $res = 1, a = 5, b = 59, c = 19;$

El cálculo del módulo podemos ir haciéndolo en cada iteración debido a las propiedades:

$5^{59}\%c = 5^{111011_2}\%c = (5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0})\%c$

$5^{59}\%c =$

$[(5^{2^5}\%c) * (5^{2^4}\%c) * (5^{2^3}\%c) * (5^{(2^2*0)}\%c) * (5^{2^1}\%c) * (5^{2^0}\%c)]\%c$

Cada término es el denominado a que iremos calculando. Ya que:

$5^{2^2}\%19 = [(5^{2^1}\%19) * (5^{2^1}\%19)]\%19$

Ejemplo exponenciación modular

Calcular $5^{59}\%19$. En primer lugar representamos en binario a 59.

$5^{59} = 5^{111011_2} = 5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0}$
Llegaremos al resultado en 6 iteraciones:
Definamos $res = 1, a = 5, b = 59, c = 19;$

El cálculo del módulo podemos ir haciéndolo en cada iteración debido a las propiedades:
$5^{59}\%c = 5^{111011_2}\%c = (5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0})\%c$
$5^{59}\%c =$
$[(5^{2^5}\%c) * (5^{2^4}\%c) * (5^{2^3}\%c) * (5^{(2^2*0)}\%c) * (5^{2^1}\%c) * (5^{2^0}\%c)]\%c$
Cada término es el denominado *a* que iremos calculando. Ya que:
$5^{2^2}\%19 = [(5^{2^1}\%19) * (5^{2^1}\%19)]\%19$

Ejemplo exponenciación modular

Calcular $5^{59}\%19$. En primer lugar representamos en binario a 59.

$5^{59} = 5^{111011_2} = 5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0}$

Llegaremos al resultado en 6 iteraciones:

Definamos $res = 1, a = 5, b = 59, c = 19;$

El cálculo del módulo podemos ir haciéndolo en cada iteración debido a las propiedades:

$5^{59}\%c = 5^{111011_2}\%c = (5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0})\%c$

$5^{59}\%c =$

$[(5^{2^5}\%c) * (5^{2^4}\%c) * (5^{2^3}\%c) * (5^{(2^2*0)}\%c) * (5^{2^1}\%c) * (5^{2^0}\%c)]\%c$

Cada término es el denominado a que iremos calculando. Ya que:

$5^{2^2}\%19 = [(5^{2^1}\%19) * (5^{2^1}\%19)]\%19$

Ejemplo exponenciación modular

Calcular $5^{59}\%19$. En primer lugar representamos en binario a 59.

$5^{59} = 5^{111011_2} = 5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0}$
Llegaremos al resultado en 6 iteraciones:
Definamos $res = 1, a = 5, b = 59, c = 19;$

El cálculo del módulo podemos ir haciéndolo en cada iteración debido a las propiedades:
$5^{59}\%c = 5^{111011_2}\%c = (5^{2^5} * 5^{2^4} * 5^{2^3} * 5^{(2^2*0)} * 5^{2^1} * 5^{2^0})\%c$
$5^{59}\%c =$
$[(5^{2^5}\%c) * (5^{2^4}\%c) * (5^{2^3}\%c) * (5^{(2^2*0)}\%c) * (5^{2^1}\%c) * (5^{2^0}\%c)]\%c$
Cada término es el denominado a que iremos calculando. Ya que:
$5^{2^2}\%19 = [(5^{2^1}\%19) * (5^{2^1}\%19)]\%19$

Ejemplo exponenciación modular

1. Dado que el dígito más de la derecha $'b'(111011)$ es **1**:

   $res = (res * a)\%c = (1 * 5)\%19 = 5$
   $a = (a * a)\%c = (5 * 5)\%19 = 6$
   $b/ = 2(b = 11101)$

2. Dado que el dígito más de la derecha $'b'(11101)$ es **1**:

   $res = (res * a)\%c = (5 * 6)\%19 = 11$
   $a = (a * a)\%c = (6 * 6)\%19 = 17$
   $b/ = 2(b = 1110)$

Ejemplo exponenciación modular

3. Dado que el dígito más de la derecha $'b'(1110)$ es **0**: No multiplicamos el valor **res** ya que no forma parte.

$$a = (a * a)\%c = (17 * 17)\%19 = 4$$
$$b/ = 2(b = 111)$$

4. Dado que el dígito más de la derecha $'b'(111)$ es **1**:

$$res = (res * a)\%c = (11 * 4)\%19 = 6$$
$$a = (a * a)\%c = (4 * 4)\%19 = 16$$
$$b/ = 2(b = 11)$$

5. Dado que el dígito más de la derecha $'b'(11)$ es **1**:

$$res = (res * a)\%c = (6 * 16)\%19 = 1$$
$$a = (a * a)\%c = (16 * 16)\%19 = 9$$
$$b/ = 2(b = 1)$$

6. Dado que el dígito más de la derecha $'b'(1)$ es **1**:

$$res = (res * a)\%c = (1 * 9)\%19 = 9$$
$$a = (a * a)\%c = (9 * 9)\%19 = 5$$
$$b/ = 2(b = 0)/$$

Ejemplo exponenciación modular

El resultado final es: 9, por lo tanto $(5^{59})\%19 = 9$ Tiempo: $\mathcal{O}(\log_2(b))$
Complexity:$\mathcal{O}(\log_2(b))$ (números de dígitos presentes ne la notación binaria del número ‘b’)
Implementation:

```
#include <cstdio>
#include <iostream>
using namespace std;

int modpowIter(int a, int b, int c) {
        int res=1;
        while(b != 0) {
                if(b%2 == 1)
                        res=(res*a)%c;

                a=(a*a)%c;
                b /= 2;
        }
        return res;
}
int main() {
        int a=5,b=59,c=19,res;
        res = modpowIter(a,b,c);
        cout << res << endl;
}
```

Contenido

Fibonacci Numbers

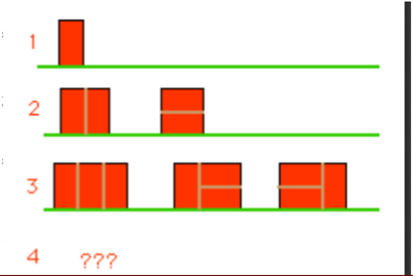
Defінción

$$fib(0) = 0, fib(1) = 1 n \geq 2, fib(n) = fib(n-1) + fib(n-2)$$

Da un patrón conocido: 0, 1, 23, 58, 13, 21, 34, 55, 89, etc.

Problema 900 - Brick Wall Patterns

If we want to build a brick wall out of the usual size of brick which has a length twice as long as its height, and if our wall is to be two units tall, we can make our wall in a number of patterns, depending on how long we want it. From the figure one observe that:...



Fibonacci Numbers

Cálculo del iésimo número

Los núemros de fibonacci son: 0,1 ,1, 2, 3, 5, 8... Partimos de una matriz y se va calculando multiplicándola por sí misma.

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

En la posición (0,0) nos queda el número iésimo de fibonacci. Si queremos calcular el k sería:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^k$$

La multiplicación es de $\mathcal{O}(\log_2(k))$ por el costo de realizar la multiplicación de matrices.

Zeckendorf's Theorem

Definición

Todo número entero positivo puede ser representado en forma única por la suma de uno o mas números diferentes Fibonacci, de forma tal que la suma no incluya dos números Fibonacci consecutivos

Contenido

¿Cuántas formas existen de modo que N elementos puedas ser tomados de K a la vez donde el orden no importa?

Ejemplo

Combinación de las letras *a,b,c,d* tomadas cada vez son:
abc, abd, acd, bcd
Son iguales a: *abc, acb, bac, bca, cab, cba*

Denominamos al número de combinaciones posibles de n elementos tomados de a r como:

$$C(N,K) = \binom{N}{K} = \frac{N!}{(N-K)!K!}$$

Dado que cada combinación de n elementos tomados de a r se calcula como *r!* permutaciones de objetos, podemos decir que: $P(n,r) = r!C(n,r)$

Coeficientes binomiales - Teorema de Newton

El coeficiente binomial $\binom{n}{k}$ es el coeficiente del término obtenido al desarrollar :

$$(x+y)^n = \sum_{i=1}^{n} \binom{n}{k} x^{n-k} y^k$$

El coeficiente del término $a^k b^{n-k}$ es C(n,k):

$$\binom{n}{k} = \frac{N!}{(N-K)!K!}$$

$$\begin{aligned}
(a+b)^0 &= 1 \\
(a+b)^1 &= a + b \\
(a+b)^2 &= a^2 + 2ab + b^2 \\
(a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3 \\
(a+b)^4 &= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 \\
(a+b)^5 &= a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5 \\
(a+b)^6 &= a^6 + 6a^5b + 15a^4b^2 + 20a^3b^3 + 15a^2b^4 + 6ab^5 + b^6
\end{aligned}$$

$$\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}$$

Lemma

Dado el coeficiente binomial $\binom{n}{n-r} = \binom{n}{r}$, podemos decir, si $a + b = n$, entonces $\binom{n}{a} = \binom{n}{b}$

Ejemplo:
$\binom{10}{7} = \frac{10*9*8*7*6*5*4}{1*2*3*4*5*6*7}$ o, $\binom{10}{3} = \frac{10*9*8}{1*2*3}$

Tener en cuenta

- $\binom{n}{0} = \frac{n!}{0!n!} = 1$
- Hay n+1 términos.
- La suma de los exponentes de a y b en cada término son n.
- Los exponentes de a decrecen de n a 0 mientras que los de b se incrementan.
- Los coeficientes de cada término $\binom{n}{k}$ donde k es el exponente de a o b.

Coeficientes de los términos

Si tenemos que calcular muchas veces todos los valores, podríamos pre-calcular para no repetir cuentas, cómo se podría hacer?

SII!! Programación Dinámica.

Casos bases:

```
C(n,0) = C(n,n)=1
C(n,k)= C(n-1,k-1)+ C(n-1,k)
```

Coeficientes de los términos

Si tenemos que calcular muchas veces todos los valores, podríamos pre-calcular para no repetir cuentas, cómo se podría hacer?

SII!! Programación Dinámica.

Casos bases:

```
C(n,0)  = C(n,n)=1
C(n,k)= C(n-1,k-1)+ C(n-1,k)
```

Coeficientes de los términos

Si tenemos que calcular muchas veces todos los valores, podríamos pre-calcular para no repetir cuentas, cómo se podría hacer?

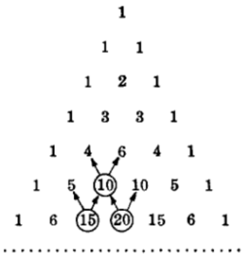
SII!! Programación Dinámica.

Casos bases:

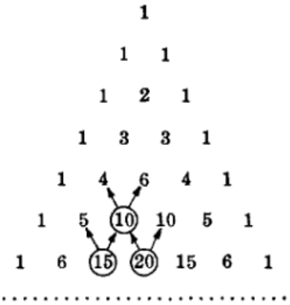
```
C(n,0) = C(n,n)=1
C(n,k)= C(n-1,k-1)+ C(n-1,k)
```

```c
#define MAXN 100 /* largest n or m */
long binomial_coefficient(n,m)
  /* calcula n tomados de a m */
{
  int i,j; /* contadores */
  long bc[MAXN][MAXN]; /* tabla de coeficientes
     binomiales */
  for (i=0; i<=n; i++) bc[i][0] = 1;
  for (j=0; j<=n; j++) bc[j][j] = 1;
  for (i=1; i<=n; i++)
    for (j=1; j<i; j++)
      bc[i][j] = bc[i-1][j-1] + bc[i-1][j];
  return( bc[n][m] );
}
```

Inicialización del triángulo de pascal.



	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0
3	1	0	0	1	0	0	0	0
4	1	0	0	0	1	0	0	0
5	1	0	0	0	0	1	0	0
6	1	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1

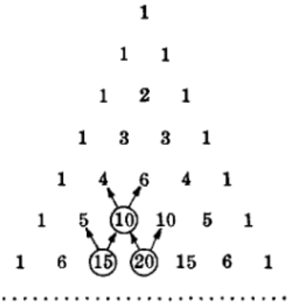


(2,1) = 2

bc[i][j] = bc[i-1][j-1] + bc[i-1][j];

bc[2][1] = bc[i-1][j-1] + bc[i-1][j] = 2;

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	1	2	1	0	0	0	0	0
3	1	0	0	1	0	0	0	0
4	1	0	0	0	1	0	0	0
5	1	0	0	0	0	1	0	0
6	1	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1

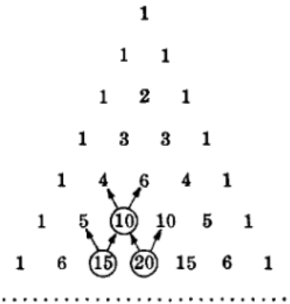


(3,1) = 3, (3,2) = 3

bc[i][j] = bc[i-1][j-1] + bc[i-1][j];

bc[3][1] = bc[i-1][j-1] + bc[i-1][j] = 3;

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	1	2	1	0	0	0	0	0
3	1	3	3	1	0	0	0	0
4	1	0	0	0	1	0	0	0
5	1	0	0	0	0	1	0	0
6	1	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1

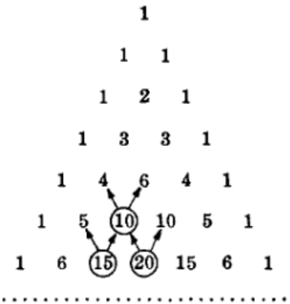


(4,1) = 4, (4,2) = 6, (4,3)=4

bc[i][j] = bc[i-1][j-1] + bc[i-1][j];

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	1	2	1	0	0	0	0	0
3	1	3	3	1	0	0	0	0
4	1	4	6	4	1	0	0	0
5	1	0	0	0	0	1	0	0
6	1	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1

Si quiero saber el **coeficiente** del término k=2, siendo n=3$\Rightarrow$ bc[3][1]



(4,1) = 4, (4,2) = 6, (4,3)=4

bc[i][j] = bc[i-1][j-1] + bc[i-1][j];

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	1	2	1	0	0	0	0	0
3	1	3	3	1	0	0	0	0
4	1	4	6	4	1	0	0	0
5	1	0	0	0	0	1	0	0
6	1	0	0	0	0	0	1	0
7	1	0	0	0	0	0	0	1

Si quiero saber el **coeficiente** del término k=2, siendo n=3 $\Rightarrow$ bc[3][1]