novática
CEPIS UPGRADE Revista de la Asociación de Técnicos de Informática

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). ***Novática*** co-edita asimismo **UP**GRADE, revista digital de **CEPIS** (*Council of European Professional Informatics Societies*), en lengua inglesa, y es miembro fundador de **UP**ENET (***UP****GRADE* **E***uropean* **NET***work*).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>
<http://www.cepis.org/upgrade>

ATI es miembro fundador de **CEPIS** (*Council of European Professional Informatics Societies*) y es representante de España en **IFIP** (*International Federation for Information Processing*); tiene un acuerdo de colaboración con **ACM** (*Association for Computing Machinery*), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AI2**, **ASTIC**, **RITSI** e **Hispalinux**, junto a la que participa en **ProInnova**.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Padilla 66, 3º, dcha., 28006 Madrid
Tlfn.914029391; fax.913093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tlfn./fax 963330392 <secreval@ati.es>
Administración y Redacción ATI Cataluña
Via Laietana 46, ppal. 1ª, 08003 Barcelona
Tlfn.934125235; fax 934127713 <secregen@ati.es>
Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza.
Tlfn./fax 976235181 <secreara@ati.es>
Redacción ATI Andalucía <secreand@ati.es>
Redacción ATI Galicia <secregal@ati.es>
Suscripción y Ventas <http://www.ati.es/novatica/interes.html>, ATI Cataluña, ATI Madrid
Publicidad
Padilla 66, 3º, dcha., 28006 Madrid
Tlnf.914029391; fax.913093685 <novatica@ati.es>
Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona.
Depósito legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVAEC
Portada: Resolución en marcha - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

Nº 211, mayo-junio 2011, año XXXVII

sumario

Tema del próximo número: "*Innovación y emprendimiento en Informática*"

Triangulo de Pascal y la Potencia Binomial

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cardenas
Laboratorio de Investigación de Software MsLabs, Dpto. Ing. en Sistemas de Información, Facultad Regional Córdoba - Universidad Tecnológica Nacional (Argentina)

<jotacastillo@gmail.com>,
<diegojserrano@gmail.com>,
<ing.marinacardenas@gmail.com>

El enunciado de este problema apareció en el número 209 de ***Novática*** (enero-febrero 2011, p. 76).

Para resolver este problema nos basaremos en la observación de que el Triángulo de Pascal puede ser convenientemente representado de la siguiente forma que se visualiza en la **figura 1**.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

Figura 1. Coeficientes binomiales representados mediante una matriz triangular izquierda.

Esta forma de representar los coeficientes binomiales constituye una manera casi directa de implementación algorítmica empleando matrices.

El programa presentado a continuación ha sido modelado utilizando el paradigma orientado a objetos. Sin embargo, dada la simplicidad del problema y su resolución netamente algorítmica, la codificación de la solución se realizó solamente en una clase Main utilizando dos métodos estáticos. Por ello, la implementación de esta solución sería muy similar a la necesaria en un lenguaje de programación como C/C++ u otros lenguajes imperativos.

La clase Main contiene el método principal "main()" mediante el cual solicitaremos el ingreso de datos por la entrada estándar. Se realiza la lectura de tres números enteros: a, b, y n (donde: n<=10, y que 0<=a, b <= 4) correspondientes a la fórmula $(a + b)^n$. De esta manera, la salida esperada del programa será el valor de $(a + b)^n$ seguido de los coeficientes binomiales del n-ésimo nivel del triángulo.

Dicho programa contiene un segundo método llamado "calcularTriangulo()" que implementa el cálculo de las 10 primeras filas (ya que n<=10) de la matriz de coeficientes binomiales (ver **figura 1**). Nótese también que, tanto la fila 0 como la diagonal de la matriz, tendrán siempre asignado el valor 1 siguiendo la ecuación de la potencia del binomio que se corresponde con elevar un número a la potencia 0. El cálculo del valor del coeficiente para la i-ésima fila se realiza en base a la suma del valor de los coeficientes de la fila inmediatamente anterior (i-1), el valor de la columna actual (j) y el de la columna anterior (j-1). Esta matriz es necesaria para poder imprimir los coeficientes binomiales del n-ésimo nivel del triángulo de Pascal.

Posteriormente, una vez construida esta matriz solo resta imprimir el valor de $(a + b)^n$ y luego los coeficientes correspondientes de ese nivel.

Para imprimir dicho valor tenemos dos alternativas:
- La primera es simplemente computar este valor ya que los valores de las variables a, b y n son conocidos.
- La segunda alternativa es seguir la definición y calcular

$$\sum_{k=0}^{n} F(n,k)a^{n-k}b^{k}$$

que es el valor de la expresión generalizada de la potencia binomial. Se proveen ambas alternativas en el código, y se comentan las líneas de código correspondientes a la segunda alternativa. Cabe destacar que entre ambas opciones es preferible la alternativa 1 ya que utiliza información conocida del problema y tiene una complejidad temporal de O(k) frente a la segunda alternativa cuya complejidad es O(n).

Finalmente, es necesario imprimir el valor de los coeficientes binomiales, para lo cual basta con recorrer la n-ésima fila de la matriz denominada "pascal" en el código fuente.

A continuación se provee el código fuente de la solución:

```
import  java.util.Scanner;

public  class  Main
{
   static  int[][]pascal  =  new  int[10][10];
   public  static  void  main(String[]  args)
   {
      calcularTriangulo();
      int a, b, n;
      Scanner  sc  =  new  Scanner(System.in);

      while  (sc.hasNextInt())
      {
            a  =  sc.nextInt();
            b  =  sc.nextInt();
            n  =  sc.nextInt();

            int  suma  =  0;

            //Alternativa  1
            suma=(int)Math.pow(a+b,n);

            //Alternativa  2:
            //for(int  k  =  0;  k  <=  n;  k++)
            //    suma  +=  (int)(pascal[n][k]  *
Math.pow(a,n-k)  *  Math.pow(b,k));

            System.out.print(suma);
            for  (int  k  =  0;  k  <=  n;  k++)
                  System.out.print("     "     +
pascal[n][k]);
            System.out.println();
      }
      return;
   }

   static  void  calcularTriangulo()
   {
      int  i,j;
      pascal[0][0]  =  1;
      for  (i  =  1;  i  <  10;  i++)
      {
            pascal[i][0]  =  1;
            for  (j  =  1;  j  <  i;  j++)
            {
                  pascal[i][ j]  =  pascal[i  -  1][j  -
1]  +  pascal[i  -  1][j];
            }
            pascal[i][j]  =  1;
      }
```