novática
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).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>

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. 963740173 <novatica_prod@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: La casa danzante - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

sumario

Nº 217, mayo-junio 2012, año XXXVIII

Tema del próximo número: "Sistemas multiagente"
(Número distribuido solamente en versión digital)

El Problema del *Superbowling*

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas
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 215 de **Novática** (enero-febrero 2012, p.60)

Este problema nos acerca al concepto de los números triangulares <http://es.wikipedia.org/wiki/Número_triangular>, que son aquellos números iguales a la sumatoria de los números naturales desde 1 hasta algún n.

De esta forma, los primeros números triangulares son:

T1 = 1
T2 = 1 + 2 = 3
T3 = 1 + 2 + 3 = 6
T4 = 1 + 2 + 3 + 4 = 10

La denominación es consecuencia de la posibilidad de organizar en forma de triangulo equilátero conjuntos de objetos cuya cardinalidad sea igual a algún número triangular. En el ejemplo del juego del *bowling*, los 10 pinos son ubicados formando un triángulo equilátero con 4 pinos por lado.

El problema nos solicita calcular cuantas hileras de pinos serán necesarias para ubicar una cierta cantidad de pinos. Eso equivale a calcular el menor número triangular que sea mayor que el dato de entrada N.

La solución más simple para obtener el resultado es calcular todos los números triangulares acumulando los números naturales hasta que el acumulador supere a N, contando las iteraciones. La consigna del problema indica que la solución siempre será menor a 105 (dado que para el caso extremo de 10^9 pinos, hacen falta 44721 hileras), por lo tanto una solución de fuerza bruta no consumirá mucho tiempo de procesamiento. Más aun, todos los números triangulares involucrados pueden ser precalculados y almacenados en un arreglo para luego realizar una búsqueda binaria sobre ellos.

La codificación de esta solución se muestra en el método ContarHilerasAcumulando().

Por otro lado se puede realizar el siguiente análisis:

$$T_n = \frac{n \times (n+1)}{2}$$

$$2T_n = n^2 + n$$

Descartando n en el lado derecho de la igualdad, se puede aproximar:

$$n \approx \sqrt{2T_n}$$

Dado que n siempre es positivo, el resultado de la última ecuación siempre será menor al real, por lo tanto el número n encontrado será igual al resultado del problema o menor. Con un análisis similar se puede concluir que el error introducido nunca será mayor a la unidad.

De esta manera podremos resolver nuestro problema calculando la cantidad de hileras necesarias aplicando la última ecuación y verificando si el resultado cumple con la consigna. Si no la cumple basta con sumar 1 al resultado. Esta solución está implementada en el método ContarHilerasEcuacion().

La solución propuesta para este problema se detalla a continuación:

```
package superbowling;

import java.util.Scanner;

public class Superbowling {

    public static void main(String[] args) {
        int C;
        Scanner sc = new Scanner(System.in);

        C = sc.nextInt();
        while (C– > 0)
        {
            int T = sc.nextInt();
            System.out.println(ContarHilerasEcuacion(T));
        }
    }

    public static int ContarHilerasSumando(int n) {
        int hilera = 0;
        int acumulador = 0;
        do {
            hilera++;
            acumulador+=hilera;
        } while (acumulador < n);
        return hilera;
    }

    public static int ContarHilerasEcuacion(int n) {
        double raiz = Math.sqrt(n*2);
        int hilera = (int)Math.floor(raiz);
        int triangular = (hilera*(hilera+1)/2);
        if (triangular < n) hilera++;
        return hilera;
    }
}
```