

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** edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (UPGRADE European Network).

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

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**, **AIZ**, **ASTIC** e **HispanLinux**, junto a la que participa en **ProInnova**.

Consejo Editorial
 Antoni Carbonell Nogueras, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Grifó, Asunción Yturbe Herranz

Coordinación Editorial
 Llorenç Pagés Casas <pages@ati.es>
Composición y autedición
 Jorge Llácer Gil de Ramates
Traducciones
 Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>, Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid
Administración
 Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores
Acceso y recuperación de la información
 José María Gómez Hidalgo (Universidad Europea de Madrid), <jmgomez@uem.es>
 Manuel J. Peña López (Universidad de Huelva), <manuel.mano@diesta.uhu.es>
Administración Pública electrónica
 Francisco López Crespo (MAE), <flc@ati.es>
 Gumersindo García Arribas (MAP), <gumersindo.garcia@map.es>
Arquitecturas
 Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>
 Jordi Tubellà Morgadas (DAC-UPC), <jordi@ac.upc.es>
Auditoría IT/ITe
 Marisa Touriño Troilito, <marinatourino@marinatourino.com>
 Manuel Palao García-Suelto (ASIA), <manuel@palao.com>
Berecho y tecnologías
 Isabel Hernando Coliázos (Fac. Derecho de Donostia, UPV), <ihernando@legalek.net>
 Elena Davara Fernández (de Marcos Davara & Davara), <edavara@davara.com>
Escadanza Universitaria de la Informática
 Joaquín Epeleta Mateo (CPS-UZAR), <epeleta@posta.unizar.es>
 Cristóbal Pareja Flores (DSIP-UCM), <cpareja@sip.ucm.es>

Entorno digital personal
 Alonso Alvarez Garcia (TID), <aag@tid.es>
 Diego Gachet Paez (Universidad Europea de Madrid), <gachet@uem.es>
Basión del Conocimiento
 Joan Baiget Solé (Cap Gemini Ernst & Young), <joan.baiget@ati.es>
Informática y Filosofía
 José Ángel Olivas Varela (Escuela Superior de Informática, UCLM)
 Karim Gherrab Martin (Indra Sistemas)
Informática Gráfica
 Miguel Chover Selles (Universitat Jaume I de Castellón), <chover@lsi.uji.es>
 Roberto Vivó Herrando (Eurographics, sección española), <rvivo@dsic.upv.es>

Ingeniería del Software
 Javier Dolado Cosin (DLSI-UPV), <dolado@si.ehu.es>
 Luis Fernández Sanz (PRIS-El-UEM), <lufern@pris.esi.uem.es>
Inteligencia Artificial
 Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV)
 <vboti.vinglada@dsic.upv.es>
Interacción Persona-Computador
 Julio Abascal González (F-UPV), <julio@sti.ehu.es>
 Jesús Lorés Vidal (Univ. de Lleida), <jesus@eup.udl.es>
Lengua e Informática
 M. del Carmen Ugarte García (IBM), <cuarte@ati.es>
Lenguajes Informáticos
 Andrés Marín López (Univ. Carlos III), <amarin@it.ucm.es>
 J. Ángel Velázquez Iturbide (ESCET-URJC), <a.velazquez@escet.urjc.es>

Lingüística computacional
 Xavier Gómez Guinovart (Univ. de Vigo), <xgg@uvigo.es>
 Manuel Palomar (Univ. de Alicante), <mpalomar@disi.ua.es>
Mundo estudiantil
 Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM), <a.vazquez@ieee.org>
 Federico G. Mon Trotti (RITSI) <gnu.fede@gmail.com>
Profesión Informática
 Rafael Fernández Calvo (ATI), <rfcalvo@ati.es>
 Miquel Sàrries Grifó (Ayto. de Barcelona), <msarries@ati.es>

Redes y servicios telemáticos
 José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>
 Javier Solé Pareta (DAC-UPC), <pareta@ac.upc.es>
Seguridad
 Javier Arellito Bertolin (Univ. de Deusto), <jarellito@eside.deusto.es>
 Javier López Muñoz (ETSI Informática-UMA), <jlm@cc.uma.es>
Sistemas de Tiempo Real
 Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <jaalonso.jpunte@dit.upm.es>
Software Libre
 Jesús M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC), <jmgonzales@gsyc.es>
Tecnología de Objetos
 Jesús García Molina (DIS-UM), <jmolina@um.es>
 Gustavo Rossi (LIFA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>
Tecnologías para la Educación
 Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>
 Juliá Minguillón i Alfonso (UOC), <jminguillona@uoc.edu>
Tecnologías y Empresa
 Didac López Butifull (Universitat de Girona), <didac.lopez@ati.es>
TIC y Turismo
 Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <aguayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
 Padilla 66, 3º dcha., 28006 Madrid
 Tfn. 91 4029391; fax. 91 3093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
 Av. del Reino de Valencia 23, 46005 Valencia
 Tfn./fax 963303932 <secreval@ati.es>
Administración y Redacción ATI Cataluña
 Ciudad de Granada 131, 08018 Barcelona
 Tfn. 934125235; fax 934127713 <secregen@ati.es>
Redacción ATI Andalucía
 Isaac Newton, s/n, Ed. Sadleir,
 Isla Cortija 41092 Sevilla, Tfn./fax 954460779 <secreand@ati.es>
Redacción ATI Aragón
 Lagasca 9, 3-B, 50006 Zaragoza.
 Tfn./fax 976235181 <secreara@ati.es>
Redacción ATI Asturias-Cantabria <gp-astucant@ati.es>
Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>
Suscripción y Ventas
 <<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid
Publicidad
 Padilla 66, 3º dcha., 28006 Madrid
 Tfn. 91 4029391; fax. 91 3093685 <novatica.publicidad@ati.es>
Imprenta
 Berria S.A., Juan de Austria 66, 08005 Barcelona.
Distribución legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVACV
Partidas: "Pioneros TIC" (Charles Babbage), RFCalvo / (C) Rafael Fernández Calvo 2006
Diseño: Fernando Agresta / © ATI 2006

editorial

El tema del canon digital no cesa en resumen > 02

La Web que cambió nuestras vidas > 02

Llorenç Pagés Casas

noticias IFIP

Creación del TC14: Comité Técnico en Computación aplicada al entretenimiento > 03

Pedro Antonio González Calero

monografía

Servicios Web

(En colaboración con UPGRADE)

Editores invitados: *José Carlos del Arco, Jesús Arias Fisteus, Óscar Corcho, Jorge Cardoso*

Presentación. Los Servicios Web desde un punto de vista empresarial y académico > 04

José Carlos del Arco, Jesús Arias Fisteus, Óscar Corcho, Jorge Cardoso

Servicios Web: introducción y estado del arte > 06

Óscar Corcho García, José Carlos del Arco Prieto, Jesús Arias Fisteus

El W3C y la estandarización en los Servicios Web > 11

Martín Álvarez Espinar

Experiencias de integración e interoperabilidad en el Entorno Sanitario > 15

Antonio García Landeira

Desarrollo dirigido por modelos de propiedades extra-funcionales para Servicios Web: un caso de estudio desde las perspectivas del servicio y del cliente > 19

Guadalupe Ortíz Bellot, Juan Hernández Núñez

Hacia un proceso automatizado de trading > 26

Pablo Fernandez Montes, Manuel Resinas Arias de Reyna, Rafael Corchuelo Gil

Servicios Web Semánticos con WSMO > 30

Holger Lausen, Jos de Bruijn, Uwe Keller, Rubén Lara

Hacia la selección semántica de servicios para integración B2B > 34

Andreas Friesen, Kioumars Namiri

Aprovechando modelos de e-commerce para el desarrollo de aplicaciones basadas en Servicios Web > 40

Abraham Nieva de la Hidalga, Liping Zhao, Pedro R. Falcone Sampaio

secciones técnicas

Ingeniería del Software

Obtención de requisitos en ingeniería Web > 45

Stephanos Mavromoustakos, Katerina Papanikolaou

Tecnologías para la Enseñanza

¿Pueden los ordenadores evaluar automáticamente preguntas abiertas? > 50

Diana Pérez Marín, Enrique Alfonso Cubero, Pilar Rodríguez Marín

Tecnologías y Empresa

Impacto de la implantación distribuida de Sistemas Integrados de Gestión > 54

Salvador Bueno Ávila, José Luis Salmerón Silvera

Referencias autorizadas > 58

sociedad de la información

Futuros emprendedores

RingCode2Job: manejando una impresora multifuncional mediante interfaz web > 64

Mikel Fernández de Matauco Amestoy, Irantzu Marquinez Nieto

Novática interactiva

Blogs: en la vanguardia de la nueva generación web > 68

Antonio Fumero, Fernando Sáez Vacas

Los Servicios Web en la práctica > 74

Foro de Debate

Programar es crear

Cuadrados (CUPCAM 2005, problema fase local UCM, solución) > 75

Luis Hernández Yáñez, Cristóbal Pareja Flores

asuntos interiores

Coordinación editorial / Fe de erratas / Socios Institucionales > 77

Cuadrados

Luis Hernández Yáñez, Cris-
tóbal Pareja Flores
Universidad Complutense de Madrid

<luis,cpareja}@sip.ucm.es>

El enunciado de este problema apareció en el número 182 de *Novática* (julio-agosto 2006, p. 74). Es uno de los problemas de la fase local de la Universidad Complutense de Madrid del III Concurso Universitario de la Comunidad Autónoma de Madrid (CUPCAM 2005) del que ATI fue entidad colaboradora.

La verdad es que se trata de un problema extremadamente sencillo, pensado en esta ocasión para la fase de entrenamiento, previa al concurso real.

Según se recordará, el enunciado de este problema pedía un programa que, partiendo de una colección de puntos del plano discreto $N \times N$, calculaba el número de cuadrados distintos que podían formarse, usando esos puntos como vértices.

La idea trivial que surge en primer lugar, comprobar todas las tuplas posibles de cuatro puntos, no es buena, ya que deben evitarse las repeticiones: el cuadrado (p, q, r, s) es el mismo que el (q, r, s, p) , y que sus otras 22 permutaciones existentes. Evitaremos las repeticiones imponiendo un orden de búsqueda de los vértices:

■ Recorremos la lista de puntos eligiendo cada uno de ellos (llamémosle p) como el primer vértice posible de la tupla. En cada

caso, los otros tres vértices se buscarán entre los que siguen, de forma que p se busca sólo entre los puntos 1 y $n - 3$.

■ Recorremos cada uno de los siguientes a p como el vértice (q) que juega el papel (único) de vértice opuesto a p por una diagonal. Debemos asegurarnos de que q sea distinto de p .

■ Los otros dos puntos del hipotético cuadrado vienen determinados de forma única para esos vértices opuestos p y q . Así pues, calculamos los puntos r y s que deberían estar en la lista de puntos y comprobamos si efectivamente están en ella, a continuación de p .

Pero como en la lista sólo tenemos puntos con coordenadas enteras, antes de buscarlos comprobaremos que r y s tengan efectivamente coordenadas enteras.

Y ya sólo quedan detalles técnicos de geometría elemental: dados p, q , tenemos:

■ $diagPQ$ es el vector que va de p a q .

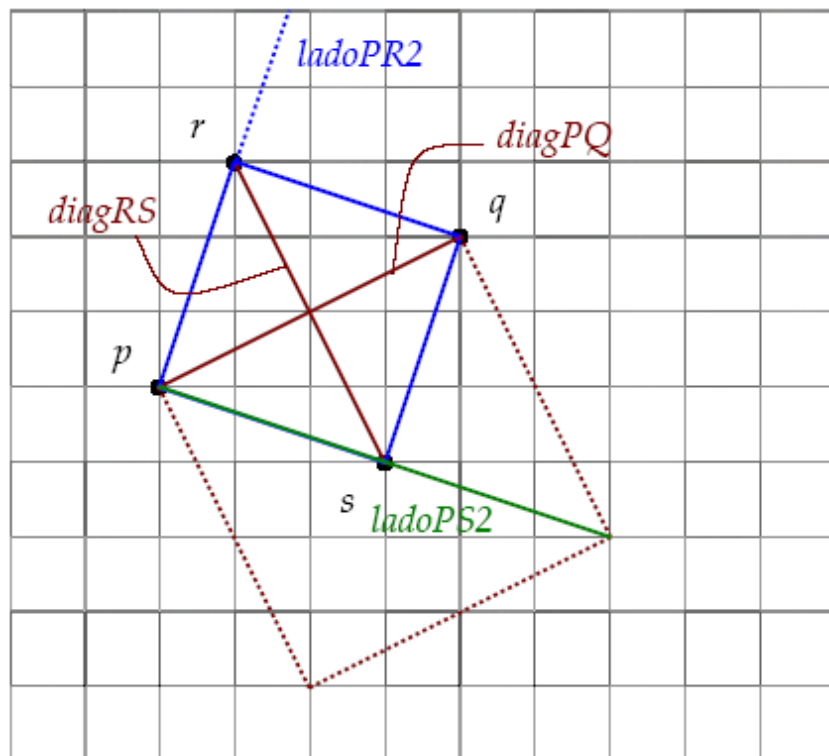
■ $diagRS$ es vector perpendicular a $diagPQ$, de igual módulo.

■ $ladoPS2$ es la suma de los vectores diagonales y el doble del lado ps . Como usaremos su mitad para llegar a s desde p , sus componentes deben ser pares para que pueda haber cuadrado.

■ $ladoPR2$ es la resta de los vectores diagonales y el doble del otro lado, pr . También sus componentes deben ser pares, aunque no será necesario comprobar esta condición, porque si las componentes de $ladoPR2$ lo son, éstas también lo serán.

■ En caso de que las paridades se tengan, sólo queda calcular los otros puntos y ver si están: $s = p + ladoPS2 / 2$ y $r = p + ladoPR2 / 2$.

La solución resulta tan breve que la incluimos completa, incluyendo la lectura de la secuencia de casos y la detección del final: el caso con cero puntos.



$$p = (2, 5)$$

$$q = (6, 7)$$

$$diagPQ = (4, 2)$$

$$diagRS = (2, -4)$$

$$ladoPS2 = (4, 2) + (2, -4) = (6, -2)$$

$$ladoPR2 = (4, 2) - (2, -4) = (2, 6)$$

$$s = (2, 5) + (6, -2) / 2 = (5, 4)$$

$$r = (2, 5) + (2, 6) / 2 = (3, 8)$$

```
#include <stdio.h>
#include <iostream>

using namespace std;

const int NumMaxPuntos = 1000;

typedef struct {
    int x, y;
} Punto;

void leerCaso (Punto listaPuntos[], int & numPuntos) {
    cin >> numPuntos;
    if (numPuntos != 0) {
        for (int i=0; i<numPuntos; i++) {
            cin >> listaPuntos[i].x >> listaPuntos[i].y;
        }
    }
}

bool distintos(Punto const p, Punto const q) {
    return p.x != q.x || p.y != q.y;
}

bool estaDesde(int izda, Punto p, Punto listaPuntos[], int numPuntos)
{
    bool encontrado = false;
    int i = izda+1;
    while (!encontrado && i < numPuntos) {
        if (p.x == listaPuntos[i].x && p.y == listaPuntos[i].y)
            encontrado = true;
        else
            i++;
    }
    return encontrado;
}

int main() {
    Punto listaPuntos[NumMaxPuntos];
    int numPuntos;
    do {
        leerCaso(listaPuntos, numPuntos);
        if (numPuntos != 0) { // conteo de cuadrados:
            int numCuadrados = 0;
            for (int i=0; i<numPuntos-3; i++) {
                for (int j=i+1; j<numPuntos; j++) {
                    // hallar los otros vértices:
                    Punto p = listaPuntos[i];
                    Punto q = listaPuntos[j];
                    Punto diagPQ = {q.x - p.x, q.y - p.y};
                    Punto diagRS = {diagPQ.y, -diagPQ.x};
                    Punto ladoPS2 = {diagPQ.x + diagRS.x,
                                       diagPQ.y + diagRS.y};
                    Punto ladoPR2 = {diagPQ.x - diagRS.x,
                                       diagPQ.y - diagRS.y};
                    Punto s = {p.x + ladoPS2.x/2, p.y + ladoPS2.y/2};
                    Punto r = {p.x + ladoPR2.x/2, p.y + ladoPR2.y/2};
                    // ver si los lados eran de coordenados enteras,
                    // y esos vértices están en la lista:
                    if (distintos(p, q) &&
                        ladoPR2.x%2 == 0 && ladoPR2.y%2 == 0 &&
                        estaDesde(i, r, listaPuntos, numPuntos) &&
                        estaDesde(i, s, listaPuntos, numPuntos)) {
                        numCuadrados++;
                        cout << p << q << endl; // sólo para comprobaciones
                    }
                }
            }
            cout << numCuadrados << endl;
        }
    } while (numPuntos != 0);
    //-----
    return 0;
} // main
```

■ También se puede anticipar la condición que examina si los puntos p y q son distintos, y la que comprueba si las coordenadas de r y s serán enteras.

Pero estas mejoras disminuyen la claridad del código, y lo cierto es además que esas mejoras son de poca monta. En un concurso de programación prima el tiempo del programador, de forma que dejamos el código como está.

El coste de examinar cada caso es cúbico con respecto al número de vértices.

Se podría reorganizar el código para lograr pequeñas mejoras:

■ Por ejemplo, sacar el cálculo de p (que no depende de j) del bucle “ j ”:

```
for (int i=0; ...
    Punto p = listaPuntos[i];
for (int j=i+1; ...
```