

**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). **Novática** edita también **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.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** y **ASTIC**.

**Consejo Editorial**  
Antoni Carbonell Noguera, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Palás Codina, Fernando Pizera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Griño, Asunción Yurbe Herranz

**Coordinación Editorial**  
Rafael Fernández Calvo <rfcalvo@ati.es>

**Composición y autoedición**  
Jorge López Gil de Ramales

**Traducción**  
Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lingua-informatica/>>

**Administración**  
Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

**Secciones Técnicas: Coordinadores**

**Administración Pública electrónica**  
Gumersindo García Arribas, Francisco López Crespo (MAP)  
<gumersindo.garcia@map.es> <flc@ati.es>

**Arquitecturas**  
Enrique F. Torres Moreno (Universidad de Zaragoza) <enrique.torres@unizar.es>  
Jordi Tubella Margadas (DAC-UPC) <jordi@dac.upc.es>

**Auditoría TIC**  
Marina Touriño Troitino, Manuel Palao García-Suelto (ASIA)  
<marinatourino@marinatourino.com>, <manuel@palao.com>

**Bases de datos**  
Coral Calero Muñoz, Mario G. Piattini Velthuis  
(Escuela Superior de Informática, UCLM)  
<Coral.Calero@uclm.es> <mpiattini@inf-cr.uclm.es>

**Borracho y tecnologías**  
Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <ihernando@legalek.net>  
Elena Davara Fernández de Marcos (Davara & Davara) <edavara@davara.com>

**Enseñanza Universitaria de la Informática**  
Joaquín Ezpeleta Mateo (CPS-UZAR) <ezpeleta@posta.unizar.es>  
Cristóbal Pareja Flores (DSP-UCM) <cpajef@isp.ucm.es>

**Gestión del conocimiento**  
Joan Baiget Solé (Cap Gemini Ernst & Young) <jbaiget@ati.es>

**Informática y Filosofía**  
Josep Corco Juvina (UIC) <jcorco@unica.edu>  
Esperanza Marcos Martínez (ESCET-URJC) <cuca@eset.urjc.es>

**Informática Gráfica**  
Miguel Chover Selles (Universitat Jaume I de Castellón) <chover@lsi.uji.es>  
Roberto Vivo Herrero (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) <lufem@pris.esi.uem.es>

**Inteligencia Artificial**  
Federico Barber Sanchis, Vicente Botti Navarro (DSIC-UPV)  
<fvotti\_barber@dsic.upv.es>

**Información Persona-Computer**  
Julio Abascal González (FI-UPV) <julio@si.ehu.es>  
Jesus Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

**Internet**  
Alonso Alvarez García (TID) <alonso@ati.es>  
Llorenç Pagès Casas (Indra) <pages@ati.es>

**Lengua e Informática**  
M. del Carmen Ugarte García (IBM) <cuarte@ati.es>

**Lenguajes Informáticos**  
Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>  
J. Angel Velázquez Turbide (ESCET-URJC) <a.velazquez@eset.urjc.es>

**Librerías e Informática**  
Alfonso Escolano (FIR-Univ. de La Laguna) <aescolan@ull.es>

**Lingüística computacional**  
Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>  
Manuel Palomar (Univ. de Alicante) <mpalomar@dsi.ua.es>

**Mundo estudiantil**  
Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM)  
<a.vazquez@iesee.org>

**Profesión Informática**  
Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>  
Miquel Sàrries Griño (Info. de Barcelona) <msarries@ati.es>

**Redes y servicios informáticos**  
Luis Guisjar Coloma (DCOM-UPV) <lguisjar@com.upv.es>  
Josep 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)  
<aalonso@ipente@dit.upm.es>

**Software Libre**  
Jesus M. González Barahona, Pedro de las Heras Quirós  
(GSYC-URJC) <jm.gonzalez@gsyc.es>

**Tecnología de Objetos**  
Jesus García Molina (DIS-UM) <jmolina@correo.um.es>  
Gustavo Rossi (LFIA-UNLP, Argentina) <gustavo@sol.info.unlp.edu.ar>

**Tecnologías para la Educación**  
Juan Manuel Dodero Beato (UC3M) <dodero@inf.uc3m.es>

**Tecnologías y Empresa**  
Pablo Hernández Medrano (Bluemat) <pablohm@bluemat.biz>

**TIC para la Sanidad**  
Valentín Masero Vargas (DI-UNEX) <vmasero@unex.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 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. 914029391, fax. 913093685 <novatica@ati.es>

**Composición, Edición y Redacción ATI Valencia**  
Av. del Reino de Valencia 23, 46005 Valencia  
Tfn./fax. 963300392 <secretal@ati.es>

**Administración y Redacción ATI Cataluña**  
Ciudad de Granada 131, 08018 Barcelona  
Tfn. 934125235; fax. 934127713 <secretgen@ati.es>

**Redacción ATI Andalucía**  
Isaac Newton, s/n, Ed. Saditel,  
Isla Cartuja 41092 Sevilla, Tfn./fax. 954460779 <secretand@ati.es>

**Redacción ATI Aragón**  
Lagasca 9, 3-8, 50006 Zaragoza  
Tfn./fax. 976235181 <secretara@ati.es>

**Redacción ATI Asturias-Cantabria** <gp-astucant@ati.es>

**Redacción ATI Castilla-La Mancha** <gp-clmancha@ati.es>

**Redacción ATI Galicia**  
Recinto Ferial s/n, 36540 Silleda (Pontevedra)  
Tfn. 986581413; fax. 986580162 <secretgal@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. 914029391, fax. 913093685 <novatica.publicidad@ati.es>

**Imprenta**  
Derra S.A., Juan de Austria 66, 08005 Barcelona.

**Deposito legal:** B 15.154-1975 - ISSN: 0211-2124; CODEN NOVAEC

**Portada:** Antonio Crespo Foix / © ATI 2005

**Diseño:** Fernando Agresta / © ATI 2005

**en resumen**

**Normalizando la seguridad ... y buscando en la Intranet de Novática** > 02

Rafael Fernández Calvo

noticias de IFIP

**Informe de ATI sobre IFIP – Actividades 2004-2005** > 03

Ramón Puigjaner Trepal

**monografía**

**Estandarización y Seguridad TIC**

(En colaboración con UPGRADE)

Editores invitados: Paloma García López, Stefanos Gritzalis, Javier López Muñoz

**Presentación. La normalización en Seguridad TIC: una tarea colectiva** > 05

internacional y multisectorial

Paloma García López, Stefanos Gritzalis, Javier López Muñoz

**¿Dónde nacen las normas voluntarias y las recomendaciones relativas** > 07

a la seguridad de la información?

Paloma García López

**CEN/ISSS y su contribución a la estandarización europea en Seguridad** > 15

de las Tecnologías de la Información

Luc Van den Berghe

**Medidas y métricas de seguridad para los Sistemas de Información** > 19

José A. Mañas Argemí

**Auditoría de Seguridad de las TI desde la perspectiva de la normalización** > 23

Marina Touriño Troitino

**Legislación, estándares y recomendaciones relativos a la firma electrónica** > 27

Josep Lluís Ferrer Gomila, Apol·lònia Martínez Nadal

**El estándar X.509 para gestión de privilegios** > 32

David Chadwick

**Estándares de seguridad de las TIC para aplicaciones en el ámbito sanitario** > 38

Spyros Kokolakis, Costas Lambrinouidakis

**secciones técnicas**

**Bases de datos**

**Calidad de Datos en aplicaciones web: un "estado del arte"** > 45

Mª Angélica Caro Gutiérrez, Coral Calero Muñoz, Ismael Caballero Muñoz-Reja,

Mario Piattini Velthuis

**Informática gráfica**

**Generación de penumbras con hardware gráfico** > 49

Pere-Pau Vázquez Alcocer, Dani Susín Acebo

**Lenguajes informáticos**

**Una arquitectura software multicapa para la integración de sistemas** > 54

Rafael Pastor Pastor, Antonio Guevara Plaza, José Luis Caro Herrero,

Andrés Aguayo Maldonado

**Redes y servicios telemáticos**

**Ping Trunking: un mecanismo de control de congestión para tráfico** > 61

agregado basado en Vegas

Sergio Herrería Alonso, Manuel Fernández Veiga, Miguel Rodríguez Pérez,

Andrés Suárez González, Cándido López García

**Referencias autorizadas** > 67

**sociedad de la información**

**Programar es crear**

**La casa más grande (CUPCAM 2005, problema B, enunciado)** > 74

Manuel Abellanas Oar

**Dominó Solitario (CUPCAM 2005, problema A, solución)** > 75

Antonio Fernández Anta

**asuntos interiores**

**Coordinación editorial / Programación de Novática** > 76

**Normas de publicación para autores / Socios Institucionales** > 77

Antonio Fernández Anta  
 Universidad Rey Juan Carlos

# Dominó Solitario

<antonio.fernandez@urjc.es>

El enunciado de este problema apareció en el número 175 de *Novática* (mayo-junio 2005, p. 75). Es el problema A de los planteados en el III Concurso Universitario de la Comunidad Autónoma de Madrid (CUPCAM 2005), del que ATI fue entidad colaboradora.

## 1. Resumen del problema

El problema presentado en el número anterior pedía decidir si era posible formar una cadena con un conjunto dado de fichas de dominó.

## 2. Planteamiento de la solución

Una primera aproximación a este problema puede hacer pensar que la única forma de resolverlo es encontrando una cadena válida y que para ello hay que probar, en el peor caso, un número muy alto de posibles cadenas y subcadenas. Una solución de este estilo seguramente llegaría a la solución, pero sería demasiado lenta como para ser aceptada como válida en el concurso.

Sin embargo, una pista en el enunciado es que no se pide dar una cadena válida, sino sólo determinar si tal cadena existe. Esto debería hacer pensar que quizá este problema sea más sencillo que el de encontrar la cadena.

Una maduración del enunciado del problema debería llevar a darse cuenta de que se puede transformar el problema en otro tipo de problema clásico en teoría de grafos: decidir si un grafo es euleriano.

## 3. Caminos eulerianos

Un camino euleriano en un grafo es un camino que recorre todas las aristas del grafo una y sólo una vez. El problema de decidir si un grafo es euleriano o no fue brillantemente resuelto por Euler (de aquí el nombre de camino euleriano) al estudiar el problema de los puentes de Königsberg. El teorema de Euler dice que un grafo contiene un camino euleriano si y sólo si es conexo y el grado de sus vértices cumple que:

- Todos los vértices tienen grado par, o
- Todos los vértices tienen grado par, excepto dos que lo tienen impar.

Dado un grafo no es complicado escribir un programa que verifique si esta condición se cumple.

## 4. Transformación

Vamos a ver, pues, cómo transformamos el problema del dominó solitario al problema de decidir si un grafo tiene un camino euleriano. Parece claro que si la condición del primer problema es usar todas las fichas y la del segundo es usar todas las aristas, nuestra transformación debe generar un grafo

donde a cada ficha le corresponda una arista.

Además, en el dominó podemos concatenar a una ficha dada solamente otra que tenga el mismo valor que el extremo libre de la primera. Por ello, debemos hacer que una arista en el grafo obtenido pueda ser seguida de otra arista que represente una ficha de igual valor en el extremo.

Todo esto es sencillo de garantizar si transformamos el conjunto de fichas de dominó en un grafo con 7 vértices, 0, 1, ..., 6, y por cada ficha (x,y) añadimos al grafo una arista entre los vértices x e y. Tras esta transformación obtendremos un grafo que tiene un camino euleriano si y sólo si el conjunto de fichas puede formar una cadena.

## 5. Programa solución

A continuación presentamos un programa que resuelve este problema como se ha descrito. El programa construye en primer lugar un grafo a partir del conjunto de fichas de dominó dado y luego usa el teorema de Euler para determinar si la cadena deseada existe.

```
#include<stdio.h>

// Degree of each node 0,...,6.
int degree[7],
// Adjacency matrix
adj[7][7];

// Checks for connectivity
int connected()
{
    int i, j, k, l,
    // Stores reachability
    reach[7];

    for(i=0; i<7; i++)
        reach[i]=0;
    // Search first used node i
    for(i=0; degree[i]==0; i++);
    // Find reachability from i
    reach[i]=1;
    // Six iterations suffice to find all
    // reachable nodes
    for(j=1; j<7; j++)
        // Add reachable in this iteration
        for(k=i; k<7; k++)
            for(l=i; l<7; l++)
                if ((reach[k]==1) &&
                    (adj[k][l]==1))
                    reach[l]=1;
    // Check that all used nodes are
```

```
reachable
for(j=i; j<7; j++)
    if (degree[j]>0 && reach[j]==0)
        return(0);
return(1);
}

int main()
{
    int p,n,i,j,k,u,v,d;

    scanf("%d\n", &p);
    // For each problem do...
    for(i=0; i<p; i++) {
        // Clear variables
        for(j=0; j<7; j++) {
            degree[j]=0;
            for (k=0;k<7;k++) adj[j][k]=0;
        }
        // Read set size
        scanf("%d\n", &n);
        // Read set and update variables
        for(j=0; j<n; j++) {
            scanf("%d %d\n", &u, &v);
            degree[u]++;
            degree[v]++;
            adj[u][v]=1;
            adj[v][u]=1;
        }
        // Check that the degree is odd
        in none
        // or exactly two nodes
        d=0;
        for(j=0; j<7; j++)
            d += degree[j] % 2;
        if ((d != 0) && (d != 2))
            printf("NO\n");
        // If so, check for connectivity
        else if (connected())
            printf("YES\n");
        else printf("NO\n");
    }
}
```

Observa que el programa no necesita realmente crear el grafo. Le basta con almacenar el grado de cada vértice (número de fichas con cada valor) para poder comprobar si éste es par o impar y una matriz de adyacencia del grafo que usa para comprobar la conectividad.