

Novática, revista fundada en 1975, es el órgano oficial de expresión y formación continua de ATI (Asociación de Técnicos de Informática)

ATI es miembro de CEPIS (*Council of European Professional Informatics Societies*) y tiene un acuerdo de colaboración con ACM (*Association for Computing Machinery*). Tiene asimismo acuerdos de vinculación o colaboración con AdaSpain, AI² y ASTIC

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

CONSEJO EDITORIAL

Antoni Carbonell Nogueras, Francisco López Crespo, Didac López Viñas, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Roberto Moya Quiles, Gloria Nistal Rosique (Presidenta del Consejo), César Pérez Chirinos, Mario Piattini Velhuis, Fernando Piera Gómez, Miquel Sàrries Grinyó, Carmen Ugarte García, Asunción Yturbe Herranz

Coordinación Editorial
 Rafael Fernández Calvo <rfoalvo@ati.es>

Composición y autoedición
 Jorge Llácer

Administración
 Tomás Brunete, María José Fernández

SECCIONES TÉCNICAS: COORDINADORES

- Arquitecturas**
 Antonio Gonzalez Colás (DAC-UPC) <antonio@ac.upc.es>
- Bases de Datos**
 Coral Calero Muñoz, Mario G. Piattini Velhuis (Escuela Superior de Informática, UCLM) <Coral.Calero@uclm.es>, <mpiatini@inf-cr.uclm.es>
- Calidad del Software**
 Juan Carlos Granja (Universidad de Granada) <jcgranja@goliat.ugr.es>
- Derecho y Tecnologías**
 Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>
- Enseñanza Universitaria de la Informática**
 Cristóbal Pareja Flores (Dep. Sistemas Informáticos y Programación-UCM) <cpareja@sip.ucm.es>
- Euro/Efecto 2000**
 Joaquín Ríos Boutin <jrios@ati.es>
- Informática Gráfica**
 Roberto Vivó (Eurographics, sección española) <rvivo@dsic.upv.es>
- Ingeniería del Software**
 Luis Fernández (PRIS-E.I./UEM) <lufern@dpris.esi.uem.es>
- Inteligencia Artificial**
 Federico Barber, Vicente Boti (DSIC-UPV) <fvboti_fbarber@dsic.upv.es>
- Interacción Persona-Computador**
 Julio Abascal González (FI-UPV) <julio@si.ehu.es>
- Internet**
 Alonso Álvarez García (TID) <alonso@ati.es>
 Llorenç Pagés Casas (Atlante) <pages@ati.es>
- Lengua e Informática**
 M. del Carmen Ugarte (IBM) <cugarte@ati.es>
- Lenguajes informáticos**
 Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>
 J. Ángel Velázquez (ESCET-URJC) <a.velazquez@escet.urjc.es>
- Librerías e Informática**
 Alfonso Escolano (FIR- Univ. de La Laguna) <aescolan@ull.es>
- Lingüística computacional**
 Xavier Gómez Guinovart (Univ. de Vigo) <xgomez@uvigo.es>
 Manuel Palomar (Univ. de Alicante) <mpalomar@dlsi.ua.es>
- Profesión informática**
 Rafael Fernández Calvo (ATI) <rfoalvo@ati.es>
 Miquel Sàrries Grinyó (Ayto. de Barcelona) <msarries@ati.es>
- Seguridad**
 Javier Areitio (Redes y Sistemas, Bilbao) <jareitio@orion.deusto.es>
- Sistemas de Tiempo Real**
 Alejandro Alonso, Juan Antonio de la Puente (DIT-UPM) <jaalonso.jpueente@diti.upm.es>
- Software Libre**
 Jesús M. González Barahona, Pedro de las Heras Quirós (GSYC, URJC) <jgib_pheras@gsyc.escet.urjc.es>
- Tecnología de Objetos**
 Esperanza Marcos (URJC) <e.marcos@escet.urjc.es>
 Gustavo Rossi (LIFIA-UNLP, Argentina) <gustavo@sol.info.unpl.edu.ar>
- Tecnologías para la Educación**
 Benita Compostela (F. CC. PP. - UCM) <benita@dial.eunet.es>
 Josep Sales Rufi (ESPIRAL) <jsales@pie.xtec.es>
- Tecnologías y Empresa**
 Pablo Hernández Medrano <phmedrano@terra.es>
- TIC para la Sanidad**
 Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. Novática permite la reproducción de todos los artículos, salvo los marcados con © o copyright, debiéndose en todo caso citar su procedencia y enviar a Novática un ejemplar de la publicación.

- Coordinación Editorial y Redacción Central (ATI Madrid)**
 Padilla 66, 3º, dcha., 28006 Madrid
 TIF914029391; fax.913093685 <novatica@ati.es>
- Composición, Edición y Redacción ATI Valencia**
 Palomino 14, 2º, 46003 Valencia
 TIF/fax.963918531 <secreval@ati.es>
- Administración, Suscripciones y Redacción ATI Cataluña**
 Via Laietana 41, 1º, 1ª, 08003 Barcelona
 TIF934125235; fax.934127713 <secregen@ati.es>
- Redacción ATI Andalucía**
 Isaac Newton, s/n, Ed. Sadiel, Isla Cartuja 41092 Sevilla
 TIF/fax.954460779 <secreand@ati.es>
- Redacción ATI Aragón**
 Lagasca 9, 3-B, 50006 Zaragoza
 TIF/fax.976235181 <secreara@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)
 TIF986581413; fax.986580162 <secregal@ati.es>

Publicidad: Padilla 66, 3º, dcha., 28006 Madrid
 TIF914029391; fax.913093685 <novatica.publicidad@ati.es>

Imprenta: 9 Impressió S.A., Junve de Austria 66, 08005 Barcelona.
Depósito Legal: B 15.154-1975
ISBN: 0211-2124; CODEN NOVAEC

Portada: Antonio Crespo Foix / © ATI 2002

SUMARIO

En resumen: Conocimiento, sabiduría y experiencia	3
<i>Rafael Fernández Calvo</i>	
Monografía: «Gestión del Conocimiento y TIC»	
(En colaboración con Informatik/Informatique y Upgrade)	
Editores invitados: <i>Xavier Alamán</i> y <i>Christopher Lueg</i>	
Presentación. Gestión del Conocimiento y Tecnologías de la Información: relaciones y perspectivas	4
<i>Christopher Lueg</i>	
Referencias útiles sobre Gestión del Conocimiento	8
<i>Xavier Alamán</i>	
Conocer es un acto humano	9
<i>Richard McDermott</i>	
Hacia la gestión distribuida del conocimiento: implicaciones directivas y tecnológicas	13
<i>Matteo Bonifacio, Paolo Bouquet, Paolo Traverso</i>	
Herramientas informáticas para la Gestión del Conocimiento: un estudio de la situación actual	20
<i>Ruth Cobos, José A. Esquivel, Xavier Alamán</i>	
Estado de la práctica de los Sistemas de Gestión del Conocimiento: resultados de un estudio empírico	27
<i>Ronald Maier</i>	
La Gestión del Conocimiento en un despacho de abogados	35
<i>Cristina Rodríguez Morcón, José Pérez García, Juan Alberto Sigüenza Pizarro</i>	
La plataforma Sintagma: de la información al conocimiento	41
<i>Joaquín Bastos Amigo</i>	
Gestión del Conocimiento: un viaje por Europa	45
<i>Fabián García Pastor</i>	
La metodología EMEC: Gestión del Conocimiento en una comunidad virtual latinoamericana	51
<i>Daniel Pimienta, Catherine Dhaussy</i>	
Secciones Técnicas	
Euro/Efecto 2000	
Transición al Euro: la experiencia del IMI	56
<i>Joaquín Ríos Boutin, Joan Tort</i>	
Referencias autorizadas	61
Sociedad de la Información	
Personal y transferible	
La Informática y la Automática en la obra de Torres Quevedo	65
<i>Antonio Vaquero</i>	
Programar es crear	
Configuración de un aeropuerto	71
<i>25º Concurso Internacional de Programación de ACM (2001): programa A</i>	
«Ancho de banda en Internet»: solución	73
<i>Juan Céspedes</i>	
Asuntos Interiores	
Coordinación editorial / Programación de Novática	76
Normas de publicación para autores / Socios Institucionales	77
Monografía del próximo número: «eXtreme Programming»	

Programar es crear

Juan Céspedes

<cespedes@KPNQwest.es>

«Ancho de banda en Internet»:
solución

El enunciado de este problema apareció en el número 154 de *Novática* (noviembre-diciembre 2001, p. 72), siendo el problema E de los planteados en el 24º Concurso Internacional de Programación de la ACM (2000)

Nota del Editor: con la solución de este problema se cierra la serie que desde el número 146 (julio-agosto de 2000) hemos venido dedicando a los problemas planteados en la final mundial del año 2000 del prestigioso Concurso Internacional de Programación de la ACM (*Association for Computer Machinery*). En esa XXIV edición participó por primera vez un equipo de estudiantes españoles, de la ETSI de Telecomunicación de la UPM (Universidad Politécnica de Madrid), compuesto por **César Sánchez Sánchez, Álvaro Martínez Echevarría y Juan Céspedes Prieto**, con **Manuel Carro**, profesor en la Facultad de Informática de la UPM, como entrenador. Este equipo logró un muy meritorio octavo puesto.

César, Álvaro y Juan, con la ayuda puntual de algún otro compañero, han presentado desde entonces en cada número los problemas planteados así como sus propias solucio-

nes comentadas. En nombre de nuestros lectores les agradecemos, con toda sinceridad, su generoso esfuerzo, que sin duda ha servido de estímulo para muchos aficionados y profesionales de este noble Arte de la Programación.

Por fortuna, acabada esta serie, el magnífico espíritu de colaboración de varios colaboradores de *Novática* nos permite iniciar una nueva con la publicación de problemas y soluciones de la XXV edición del mismo Concurso, celebrada el año 2001, como pueden comprobar en las páginas anteriores.

La solución de este problema se basa en escribir una implementación sencilla de *maximum flow*.

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

/* Aquí guardaremos nuestro grafo con el ancho de banda entre cada par de nodos
 * /
#define MAXNODES 101
int grafo[ MAXNODES][ MAXNODES] ;

/* La siguiente rutina implementa una cola FIFO que se usaremos más adelante en
 * bfs()
 *
 * La implementación es muy sencilla; simplemente va dejando los valores de un
 * array circular usando dos variables como índices de la cabeza y la cola de las
 * posiciones ocupadas
 * /
#define MAXQUEUE 200
#define INIT 0
#define QUEUE 1
#define DEQUEUE 2
typedef struct {
    int head;
    int tail;
    int q[ MAXQUEUE] ;
} t_queue;

int
queue(int command, int *v, t_queue *q) {
    int i;

    switch (command) {
        case INIT:
            memset(q,0,sizeof(t_queue));
```

```

        break;
    case QUEUE:
        i=(q->head+MAXQUEUE-1)%MAXQUEUE;
        if (i==q->tail) {
            return 0;
        }
        q->q[ q->head] =*v;
        q->head=i;
        break;
    case DEQUEUE:
        if (q->tail==q->head) {
            return 0;
        }
        *v=q->q[ q->tail];
        q->tail=(q->tail+MAXQUEUE-1)%MAXQUEUE;
        break;
    }
    return 1;
}

```

/* Implementación de «Breadth First Search», o búsqueda en anchura para ir
* visitando todos los nodos de un grafo comenzando por las adyacencias más
* cercanas, para obtener el camino más corto en número de saltos hasta el
* destino

```

*/
void
bfs(int graph[ MAXNODES][ MAXNODES], int n, int src, int *pred) {
    char color[ MAXNODES];
    int i,v;
    t_queue q;

    for (i=0;i<n;i++) {
        pred[ i]=-1;
        color[ i]=0;
    }
    color[ src]=1;
    queue(INIT,NULL,&q);

    queue(QUEUE,&src,&q);

    while (1) {
        if (!queue(DEQUEUE,&v,&q)) {
            break;
        }

        for (i=0;i<n;i++) {
            if (graph[ v][ i] ) {
                if (color[ i]==0) {
                    color[ i]=1;
                    pred[ i]=v;
                    queue(QUEUE,&i,&q);
                }
            }
        }
        color[ v]=2;
    }
}

```

/* Y nuestra estrella: el maximum flow. Va haciendo llamadas a bfs() para
* encontrar el camino más corto desde el origen hasta el destino, calcula cuánto
* tráfico se puede meter por ese camino, decrementa el ancho de banda disponible
* en cada tramo de ese camino y vuelve a empezar para buscar otro posible
* camino, hasta que no queda ninguno

*/

```

#define INFINITY 1000000000
int
maxflow(int graph[ MAXNODES][ MAXNODES] , int n, int src, int dest) {
    int residual[ MAXNODES][ MAXNODES] ;
    int pred[ MAXNODES] ;
    int f;
    int nodo;
    int result = 0;

    memcpy(residual, graph, sizeof(residual));

    while (1) {
        bfs(residual,n,src,pred);
        if (pred[ dest] ==-1) {
            break;
        }

        nodo=dest;
        f=INFINITY;
        while (nodo!=src) {
            if (residual[ pred[ nodo]][ nodo]<f) {
                f=residual[ pred[ nodo]][ nodo] ;
            }
            nodo=pred[ nodo] ;
        }

        result += f;
        nodo=dest;
        while (nodo!=src) {
            residual[ pred[ nodo]][ nodo] -= f;
            residual[ nodo][ pred[ nodo]] -= f;
            nodo=pred[ nodo] ;
        }
    }
    return result;
}

/* Programa principal. Lee los datos de entrada, hace llamadas a maxflow() y
 * escribe la salida
 */
int
main(void) {
    int i;
    int n;
    int s,t,c;
    int networks = 0;

    while(1) {
        memset(grafo, 0, sizeof(grafo));
        scanf("%d", &n);
        if (n==0) {
            exit(0);
        }
        scanf(«%d %d %d», &s, &t, &c);
        s--; t--;
        for(i=0; i<c; i++) {
            int x,y,z;
            scanf(«%d %d %d», &x, &y, &z);
            x--; y--;
            grafo[ x][ y] = grafo[ y][ x] = z;
        }
        printf("Network %d\n", ++networks);
        printf("The bandwidth is %d.\n\n", maxflow(grafo, n, s, t));
    }
}

```