

Novática, revista fundada en 1975, 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.

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

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, AF y ASTIC

CONSEJO EDITORIAL

Antoni Carbonell Noguera, Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Roberto Moya Quiles, César Pérez Chirinos, Mario Piattini Velthuis, Fernando Pierra Gómez (Presidente del Consejo), Miquel Sarries Grinó, Carmen Ugarte García, Asunción Yturbe Herranz

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

Composición y autoedición
Jorge Llácer

Administración
Tomás Brunete, María José Fernández, Joaquín Navajas, Felicidad López

SECCIONES TÉCNICAS: COORDINADORES

Arquitecturas
Jordi Tubella (DAC-UPC) <jordit@ac.upc.es>

Bases de Datos
Coral Calero Muñoz, Mario G. Piattini Velthuis (Escuela Superior de Informática, UCLM) <Coral.Calero@uclm.es>, <mpiattini@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
Joaquín Ezpeleta (CPS-UZAR) <ezpeleta@posta.unizar.es>
Cristóbal Pareja Flores (DSIP-UCM) <cpareja@sip.ucm.es>

Informática Gráfica
Roberto Vivó (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software
Luis Fernández (PRIS-ELIUEM) <lufern@dpris.esi.uem.es>

Inteligencia Artificial
Federico Barber, Vicente Botti (DSIC-UPV) <fvbotti_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 (Indra) <spages@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>

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

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

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

Profesión informática
Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>
Miquel Sarries Grinó (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.jp puente@dit.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 (E. CC. PP. UCM) <benita@diel.unet.es>
Josep Sales Rufi (ESPIRAL) <jsales@pir.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
Tf: 914029391; fax: 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia
Palomino 14, 2º, 46003 Valencia
Tf: fax 963918531 <secreval@ati.es>

Administración y Redacción ATI Cataluña
Via Laietana 41, 1º, 08003 Barcelona
Tf: 934125235; fax 934127713 <secregen@ati.es>

Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadiel, Isla Cartuja 41092 Sevilla
Tf: fax 954460779 <secreand@ati.es>

Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza
Tf: 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)
Tf: 986581413; fax 986580162 <secregal@ati.es>

Suscripción y Ventas: <<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña y ATI Madrid

Publicidad: Padilla 66, 3º, dcha., 28006 Madrid
Tf: 914029391; fax: 913093685 <novatica.publicidad@ati.es>

Imprenta: 9-Impressió S.A., Juan de Austria 66, 08005 Barcelona.
Depósito Legal: B 15.154-1975

ISBN: 0211-2124; CODEN NOVAEC

Portada: Antonio Crespo Foix / © ATI 2002

NOVÁTICA

CEPIS UPGRADE

Revista
de la Asociación
de Técnicos
de Informática

SEPTIEMBRE - OCTUBRE 2002

159

SUMARIO

En resumen: **La Inteligencia Artificial o el sueño de Turing** 3
Rafael Fernández Calvo

Monografía: «Inteligencia Artificial: una tecnología con futuro»
(En colaboración con **Upgrade**)

Editores invitados: *Federico Barber, Vicente J. Botti y Jana Koehler*
Presentación. IA: pasado, presente y futuro 4
(Incluye «Referencias útiles sobre IA»)

Federico Barber, Vicente Botti, Jana Koehler
La comunicación oral con los computadores 8

Francisco Casacuberta Nolla
Avances en investigación sobre planificación en

Inteligencia Artificial y sus aplicaciones 11
Derek Long, Maria Fox

Tendencias en Aprendizaje Automático 25
Ramon López de Mántaras Badía

Sistemas Basados en Conocimiento 31
José Mira Mira, Ana E. Delgado García

Robots físicos cooperativos y fútbol robótico 38
Bernhard Nebel, Markus Jäger

Inteligencia Artificial y Educación: una visión panorámica 44
Maite Urretavizcaya Loinaz, Isabel Fernández de Castro

Secciones Técnicas

Bases de Datos
Metodologías de desarrollo de Sistemas de Información en la Web y análisis comparativo 49
M. José Escalona, Manuel Mejías, Jesús Torres

Enseñanza Universitaria de la Informática
Enfoques en el estudio de las interfaces de usuario 60
Juan Falgueras, Antonio Luis Carrillo, Antonio Guevara

Seguridad
La seguridad en las transacciones electrónicas a través de Internet 65
Ana Belén Alonso Conde, Rafael Moreno Vozmediano

Referencias autorizadas 70

Sociedad de la Información

Programar es crear
Gestión de una partición fija de memoria 73
25º Concurso Internacional de Programación ACM (2001): problema G
No taléis el bosque por culpa de los árboles: solución 74
Ángel Herranz, Julio Mariño, Manuel Carro, Pablo Sánchez

Asuntos Interiores

Coordinación editorial / Programación de Novática 78
Normas de publicación para autores / Socios Institucionales 79

Monografía del próximo número:
«Seguridad en Comercio/Negocio Electrónico»

Programar es crear

Traducción: Cristóbal Pareja Flores

<cpareja@sip.ucm.es>

Una técnica usada en los primeros sistemas operativos con multiprogramación involucraba la partición de la memoria principal disponible en un número de regiones, cada una con un tamaño fijo, donde distintas regiones podían tener distintos tamaños. Consideramos que la suma de los tamaños de todas las regiones es igual al tamaño de la memoria principal.

Dado un conjunto de programas, era tarea del sistema operativo el asignarlos a distintas regiones de memoria, para que pudieran ejecutarse concurrentemente. La dificultad de esto se debía al hecho de que el tiempo de ejecución de un programa podía depender de la cantidad de memoria de la que dicho programa dispusiera. Cada programa requiere un mínimo de espacio, pero si se le asigna una región mayor, su tiempo de ejecución puede prolongarse o reducirse.

La misión de este problema es determinar las asignaciones óptimas de programas a regiones de memoria. Tu programa recibe como datos los tamaños de memoria de las regiones disponibles para la ejecución de los programas y, para cada programa, una descripción de cómo depende su tiempo de ejecución de la memoria disponible para él. Tu programa debe encontrar la planificación de ejecución de los programas que minimice el lapso de tiempo medio empleado por los programas. Una planificación de ejecución es una asignación de los programas a las regiones de memoria y a tiempos, de forma que dos programas nunca usen la misma región de memoria a la vez y ningún programa sea asignado a una región de memoria de menor tamaño que su mínimo requerido. El lapso de tiempo que pasa un programa en el sistema es la diferencia entre el momento en que el programa fue enviado para su ejecución (que es cero para todos los programas de este problema) y aquél en que el programa completa su ejecución.

Entrada

La entrada de datos contendrá múltiples casos de prueba. Cada caso de prueba empieza con una línea que contiene un par de enteros m y n . El número m indica el número de regiones en que se ha repartido la memoria principal ($1 \leq m \leq 10$) y n indica el número de programas que se han de ejecutar ($1 \leq n \leq 50$).

La siguiente línea contiene m enteros positivos que expresan los tamaños de las m regiones de memoria. A continuación hay n líneas, que expresan los compromisos de espacio y tiempo para cada uno de los n programas. Cada línea empieza con un entero positivo ($k \leq 10$), seguido de k pares de enteros positivos $s_1, t_1, s_2, t_2, \dots, s_k, t_k$, tales que $s_i < s_{i+1}$

Gestión de una partición fija de memoria

25° Concurso Internacional de Programación ACM (2001): problema G

para $1 \leq i < k$. El mínimo espacio requerido por un programa es s_i , es decir, no puede ejecutarse en una región de menor tamaño que ese número. Si el programa se ejecuta en una partición de tamaño s , donde $s_i \leq s \leq s_{i+1}$ para algún i , su tiempo de ejecución será t_i . Finalmente, si el programa se ejecuta en una partición de tamaño s_k o mayor, su tiempo de ejecución será t_k .

Un par de ceros seguirá tras la entrada del último caso de prueba.

Podemos asumir que cada programa se ejecutará precisamente en el tiempo especificado para el tamaño de región dado, al margen de cuántos programas más haya en el sistema. Ningún programa requerirá más memoria que el tamaño de la región mayor.

Salida

Para cada caso de prueba, se mostrará su número (empezando en 1 y avanzando secuencialmente). Luego se escribirá el menor tiempo medio para el conjunto de programas con dos cifras decimales. A esto le sigue la descripción de una planificación de ejecución que da lugar a este lapso de tiempo medio. Se ha de dar una línea para cada programa, en el orden en que se dieron en la entrada, y que identifica el número del programa, la región en que se ejecutó (numerada según apareció en la entrada), el momento en que el programa empezó a ejecutarse y el momento en que el programa finalizó su ejecución. Se debe seguir el formato mostrado en el ejemplo de salida y poner una línea en blanco tras cada caso de prueba.

Si hay múltiples ordenaciones o asignaciones a regiones de memoria que generan el mismo lapso de tiempo medio, se dará una cualquiera de dichas planificaciones.

Nota: la solución comentada aparecerá en el próximo número de Novática

Ejemplo de Entrada

```
2 4
40 60
1 35 4
1 30 3
1 40 10
1 60 7
3 5
10 20 30
2 10 50 12 30
2 10 100 20 25
1 25 19
1 19 41
3 10 18 30 43
0 0
```

Salida para el ejemplo de entrada

```
Caso 1
Lapso de tiempo medio = 7.75
El programa 1 se ejecuta en la región 1 de 0 a 4
El programa 2 se ejecuta en la región 3 de 0 a 3
El programa 3 se ejecuta en la región 1 de 4 a 14
El programa 4 se ejecuta en la región 2 de 3 a 10

Caso 2
Lapso de tiempo medio = 35.40
El programa 1 se ejecuta en la región 2 de 25 a 55
El programa 2 se ejecuta en la región 2 de 0 a 25
El programa 3 se ejecuta en la región 3 de 0 a 19
El programa 4 se ejecuta en la región 3 de 19 a 60
El programa 5 se ejecuta en la región 1 de 0 a 18
```