

Una competición de algoritmos y estructuras de datos

M. J. Castro[†], F. Marqués[†], J. A. Sánchez[†], L. Blesa, J. V. Salas

[†]Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
(mcastro, pmarques, jandreu)@dsic.upv.es

Resumen

La asignatura “Algoritmos y estructuras de datos III” se imparte en segundo curso de la Escuela y Facultad de Informática de la Universidad Politécnica de Valencia. La asignatura tiene una carga lectiva de seis créditos, que se dividen en cuatro de teoría y problemas y dos de prácticas en el laboratorio. Las prácticas de la asignatura son fundamentales para la consecución de los objetivos de la asignatura. En esta línea, adicionalmente a la propuesta de prácticas de la asignatura (estrechamente relacionadas con los contenidos teóricos), se ha puesto en marcha este curso académico un Concurso de algoritmos con el fin de incentivar al alumnado en la realización de las prácticas.

En esta comunicación se describirá la aplicación que se planteó en el concurso, las herramientas implementadas para el seguimiento, el contraste de resultados y las medidas de tiempos puestas a disposición de los alumnos, así como los resultados conseguidos.

1 Introducción

La asignatura “Algoritmos y estructuras de datos III” se imparte en segundo curso de la Escuela y Facultad de Informática de la Universidad Politécnica de Valencia. Esta asignatura supone la continuación de los estudios de análisis y diseño de algoritmos y estructuras de datos que los alumnos reciben en primer curso con las asignaturas “Introducción a la programación”, “Algoritmos y estructuras de datos I” y “Algoritmos y estructuras de datos II”. En particular, es la última asignatura sobre este tema que los alumnos cursan en la Escuela; los alumnos de la Facultad de Informática continuarán los estudios

con la asignatura “Algorítmica”, en tercero.

2 La asignatura Algoritmos y estructuras de datos III

En la asignatura “Algoritmos y estructuras de datos III” se estudian metodologías de resolución de problemas mediante esquemas algorítmicos y su interrelación con las estructuras de datos que manejan. Los objetivos específicos de la asignatura son:

- Introducir el estudio de técnicas de diseño de algoritmos: divide y vencerás y estrategia voraz.
- Introducir el estudio de estructuras de datos básicas y avanzadas: tablas de dispersión, montículos, árboles de búsqueda, B-árboles, estructuras para manipulación de conjuntos disjuntos, grafos.
- Capacitar al alumno para seleccionar los tipos de datos y las implementaciones que resulten más convenientes para diseñar algoritmos eficientes.
- Capacitar al alumno para analizar el uso de los recursos y establecer compromisos apropiados entre complejidad temporal y espacial.
- Avanzar en el estudio de problemas algorítmicos clásicos: ordenación, búsqueda de la mediana, representación y uso de colas de prioridad y de diccionarios, exploración en grafos, etc.

La asignatura tiene una carga lectiva de seis créditos, cuatro dedicados a teoría y problemas

y dos a prácticas en el laboratorio. En la tabla 1 se presenta el programa de la asignatura resumido; junto con cada tema se señala la dedicación aproximada en horas.

Como bibliografía general del curso se recomiendan los siguientes textos: *Estructuras de Datos y Algoritmos*, de Aho, Hopcroft y Ullman [1]; *Fundamentos de Algoritmia*, de Brassard y Bratley [2]; *Introduction to Algorithms*, de Cormen, Leiserson y Rivest [3]; y *Estructuras de Datos y Algoritmos*, de Weiss [4]. El libro básico para la asignatura es *Introduction to Algorithms*, de Cormen, Leiserson y Rivest [3]. Este texto es un excelente tratado de algoritmica y se aconseja vehementemente su utilización. Para incentivar a los alumnos a que consulten el texto, se ha realizado un esfuerzo entre los profesores de la asignatura para ajustarse, en la medida de lo posible, en los aspectos de notación y ejemplos presentados, al libro. Este texto incluye todos los temas que se presentan en la asignatura, además de otros ya vistos en las asignaturas previas de estudio de algoritmos y estructuras de datos, sirviendo también para la asignatura "Algoritmica" de tercero.

La bibliografía que se maneja para las prácticas de la asignatura coincide plenamente con la utilizada en teoría, pudiéndose complementar con alguna bibliografía más específica en cada práctica.

3 Un concurso de algoritmos

Las prácticas de la asignatura son fundamentales para la consecución de los objetivos de la asignatura. En esta línea, adicionalmente a la propuesta de prácticas de la asignatura, se ha puesto en marcha este curso académico un *Concurso de algoritmos* que además pretende incentivar al alumnado en la realización de las prácticas. El objetivo principal del concurso es:

Fomentar la idea de que en la solución algorítmica a un determinado problema se debe prestar especial atención a aspectos tales como la eficiencia y la capacidad de resolver problemas de gran talla.

La experiencia de este concurso ha sido muy enriquecedora, y los trabajos presentados por los alumnos han alcanzado un nivel muy por encima

de las expectativas del profesorado. Se establecieron dos tipos de premios que incluían un incremento de la nota sobre la calificación del examen y una dotación económica aportada por la Escuela Universitaria de Informática de la Universidad.

3.1 Bases del concurso

Las bases para la participación en el concurso fueron hechas públicas a los alumnos a principios del cuatrimestre en el cual se impartía la asignatura. De esta forma los alumnos que tenían intención de participar podían tratar de incorporar los conocimientos impartidos gradualmente a lo largo del curso a la resolución del problema. Las bases, en líneas generales, eran las siguientes:

- a) Podía participar cualquier alumno matriculado en la asignatura de forma individual o por grupos.
- b) Los premios eran de dos tipos: incremento de la calificación en la asignatura, y dotación económica en metálico o en material informático o escolar. La parte económica fue subvencionada por la Escuela Universitaria de Informática.
- c) Se especificó un problema concreto a resolver y se proporcionó un conjunto de datos de prueba de dicho problema, junto con la solución para dichos datos. Otro conjunto de datos (de gran talla) se reservó inaccesible a los participantes para evaluación de las soluciones algorítmicas propuestas. No obstante, los participantes tuvieron acceso regular (aunque limitado) a un "Oráculo" que aceptaba como entrada un programa ejecutable y daba como resultado un par (C, T) , donde C era $\{SI, NO\}$ indicando la corrección o no del resultado de ejecutar el programa sobre el conjunto de evaluación y, si $C=SI$, T era el tiempo invertido por el programa en la resolución del problema para dicho conjunto. El Oráculo funcionaba siempre sobre el mismo computador y sistema operativo, cuyas características fueron divulgadas.
- d) La evaluación se realizó de manera automática, según el método del Oráculo indicado anteriormente. Los participantes que

Tabla 1: Programa de la asignatura “Algoritmos y estructuras de datos III” (AD3).

AD3	Asignatura obligatoria. Segundo curso, primer semestre 6 créditos (4 teoría y problemas + 2 prácticas) Prerrequisitos recomendados: “Introducción a la programación”, “Algoritmos y estructuras de datos I” y “Algoritmos y estructuras de datos II”	
Tema 1:	Divide y vencerás	8 horas
Tema 2:	Conjuntos	14 horas
Tema 3:	Estrategia voraz	6 horas
Tema 4:	Grafos. Algoritmos de manipulación	12 horas
Práctica 1:	Estudio del algoritmo de ordenación Quicksort	6 horas
Práctica 2:	El árbol binario de búsqueda, una estructura de datos no lineal. Implementación y estudio	4 horas
Práctica 3:	Tablas de dispersión. Implementación y estudio	5 horas
Práctica 4:	Grafos. Implementación y estudio	5 horas

deseaban concursar entregaron además las versiones definitivas de sus programas, junto con una breve memoria descriptiva, antes de una fecha límite. Los resultados obtenidos fueron juzgados por una comisión en la que participaban dos profesores de la asignatura y un miembro de la dirección de la Escuela Universitaria de Informática. El criterio prioritario era el tiempo de ejecución; como criterio secundario para decidir en caso de empates o tiempos muy similares se tuvo en cuenta la calidad de presentación, legibilidad y estructuración del código fuente del programa evaluado.

- e) Los ganadores de la competición tuvieron oportunidad de presentar los detalles de sus soluciones en una sesión pública que tuvo lugar tras el fallo de la comisión.

3.2 Planteamiento del concurso

El planteamiento de problema a resolver se hizo público al mismo tiempo que la bases de participación. El problema a resolver era el siguiente: se proporcionaban dos ficheros de caracteres ASCII que contenían texto (en castellano), incluyendo blancos y secuencias de caracteres no blancos que correspondían a *palabras*. El problema que se planteaba era obtener de manera eficiente la intersección de los vocabularios (conjunto de palabras) de los dos ficheros, es decir, el conjunto de palabras comunes a los dos textos.

Este problema, y otros problemas directamente relacionados con él, tienen interés práctico co-

mo ayuda al análisis de grandes volúmenes de texto. Por ejemplo, el complementario de la intersección (las palabras no comunes) de dos ficheros obtenidos por partición equilibrada de un texto de gran talla, contiene palabras “poco comunes”, que suelen corresponder a barbarismos, palabras extranjeras, neologismos o, simplemente errores tipográficos. Por estas razones, el problema sólo tiene interés para grandes conjuntos de datos.

Para la solución al problema propuesto se debía implementar un programa que tomase como argumentos dos ficheros de texto y produjese como resultado un fichero de texto. Para la implementación podía usarse cualquier (combinación de) lenguaje(s) de programación.

Para la evaluación de la eficiencia de las distintas soluciones algorítmicas propuestas por los concursantes, se establecieron dos ficheros de texto cuya talla total era mayor de cierto número de Mbytes. Estos datos permanecieron inaccesibles a los concursantes hasta después del fallo del concurso. Una muestra representativa de estos datos, de talla mucho más pequeña, estaba accesible a los concursantes.

Para la evaluación de los programas se disponía de un Oráculo que tenía acceso a los datos reales de la prueba final y que podía ser consultado un máximo de dos veces diarias.

El Oráculo ejecutaba los programas-consulta siempre sobre el mismo computador (HP 9000/887 en este caso) y siempre bajo el mismo sistema operativo (HP-UX versión B.10.20-C en este caso). Los concursantes tenían acce-

so sin restricciones a un computador de características similares. Además, se limitaron los recursos computacionales de manera explícita (en este caso, la memoria residente a 16384 Kbytes, el segmento de datos a 8192 Kbytes, la pila de llamadas a 4096 Kbytes, los ficheros en disco a 1024 Kbytes y tiempo de ejecución a 300 segundos). El mecanismo de acceso al Oráculo era mediante correo electrónico.

Con el fin de no saturar el Oráculo se indicó que sólo se debía acceder a él cuando se tenía una certeza razonable de que el programa a evaluar funcionaba correctamente con los datos "oficiales" de prueba. Siguiendo esta idea general, el Oráculo no suministraba resultados que pudieran servir para depuración de los programas.

3.3 Resultados

Las soluciones aportadas por los diferentes concursantes hacían uso de estructuras de datos y algoritmos estudiados en las prácticas de la asignatura. La mayoría utilizó estructuras combinadas de tablas de dispersión y árboles binarios de búsqueda. Los detalles de la solución ganadora (cuyos autores son Lucía Blesa y José Vicente Sala) se pueden consultar en el Apéndice 1. El tiempo alcanzado por esta solución era de 4.80 segundos, menos de la mitad del tiempo empleado por la siguiente mejor solución. Una cota inferior que obtuvimos para este problema, que refleja la excelencia del resultado obtenido por los ganadores, consistía en contar las palabras de ambos ficheros utilizando comandos del sistema requería un tiempo de 3.46 segundos.

4 Conclusiones

Desde el punto de vista de la consecución de los objetivos inicialmente planteados, el concurso fue positivo. Los alumnos participantes se encararon con el problema propuesto, obteniendo resultados que, a tenor de la eficiencia de la solución obtenida, podemos calificar en algún caso de excelentes.

El esfuerzo organizativo que exigió la realización del concurso ha sido muy enriquecedor porque ha supuesto afrontar tareas tales como la realización de software robusto para la medida de los algoritmos, la preparación de conjuntos de datos y la discusión de hipótesis acerca del

comportamiento de distintas soluciones.

La participación en el concurso fue menor de la deseada. Pensamos que esto fue debido a distintos factores tanto propios como ajenos al concurso. Entre ellos podemos citar, como factor propio, la dificultad aparente del problema propuesto y, como factor ajeno, la sobrecarga lectiva padecida por los alumnos. Creemos que muchos de los factores propios pueden ser eliminados o al menos atenuados en ediciones próximas del concurso.

Pensamos que nuestra experiencia puede ser puesta en práctica por otras asignaturas, o en otros ámbitos, especialmente en aquellos donde el imperativo de los contenidos dificulte el enfrentamiento con problemas de talla considerable, y donde para la medida de la adecuación de las soluciones primen aspectos cuantificables. Este tipo de problemas son muy habituales en la práctica en muchos dominios.

Agradecimientos

Deseamos expresar nuestro reconocimiento al resto de profesores de la asignatura: Francisco Casacuberta, Asunción Casanova, Natividad Prieto y especialmente a Enrique Vidal, por el desarrollo del Oráculo para la evaluación y medición de los programas.

Apéndice 1: Descripción de la solución ganadora

La idea de la solución ganadora consistía en insertar las palabras de un fichero en una estructura de datos apropiada y posteriormente eliminar de ella las palabras que aparecían en el otro fichero, escribiéndola entonces por la salida estándar. Esta estrategia tenía la ventaja de que la talla de la estructura iba reduciéndose progresivamente.

La estructura de datos utilizada consistía básicamente en una tabla de dispersión en la que las palabras se dispersaban por tres criterios:

1. Primeramente, las palabras se dispersaban de acuerdo con su longitud.
2. A continuación se dispersaban en función del código ASCII del primer carácter de la palabra.

3. En tercer lugar se utilizaba una función de dispersión consistente en sumar los valores ASCII de los caracteres de la cadena.

Finalmente las listas de colisiones se gestionaban como árboles binarios de búsqueda. La función de dispersión del tercer punto aunque no ofrecía una buena dispersión, se implementó de forma que fuese muy rápida de calcular, y en combinación con la dispersión de los dos puntos anteriores hacia que en general se obtuvieran buenos resultados.

La implementación incluía detalles tales como:

- No utilizar algoritmos recursivos, sino iterativos.
- Implementar una función propia de comparación de cadenas que aprovechase las características de la máquina.
- Hacer una gestión propia de la memoria dinámica.
- Gestionar directamente el “buffering” para optimizar la entrada/salida.

Referencias

- [1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *Estructuras de Datos y Algoritmos*. Addison-Wesley Iberoamericana, 1988.
- [2] G. Brassard and P. Bratley. *Fundamentos de Algoritmia*. Prentice-Hall, 1997.
- [3] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [4] Mark Allen Weiss. *Estructuras de datos y algoritmos*. Addison-Wesley Iberoamericana, 1995.