

UNIVERSIDAD DE GRANADA
E.T.S. DE INGENIERÍAS INFORMÁTICA y DE
TELECOMUNICACIÓN



**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

Algorítmica

Guión de Prácticas

Práctica 2: Algoritmos Divide y Vencerás

Curso 2015-2016

Grado en Informática

1. Objetivo

El objetivo de esta práctica es que el estudiante aprecie la utilidad de la técnica “divide y vencerás” para resolver problemas de forma más eficiente que otras alternativas más sencillas o directas. Para ello cada equipo de estudiantes deberá resolver uno de los problemas (escogido al azar) que se detallan más adelante, así como exponer y defender su propuesta en clase.

2. Comparación de preferencias

Muchos sitios web intentan comparar las preferencias de dos usuarios para realizar sugerencias a partir de las preferencias de usuarios con gustos similares a los nuestros. Dado un ranking de n productos (p.ej. películas) mediante el cual los usuarios indicamos nuestras preferencias, un algoritmo puede medir la similitud de nuestras preferencias contando el número de inversiones: dos productos i y j están “invertidos” en las preferencias de A y B si el usuario A prefiere el producto i antes que el j , mientras que el usuario B prefiere el producto j antes que el i . Esto es, cuantas menos inversiones existan entre dos rankings, más similares serán las preferencias de los usuarios representados por esos rankings.

Por simplicidad podemos suponer que los productos se pueden identificar mediante enteros $1, \dots, n$, y que uno de los rankings siempre es $1, \dots, n$ (si no fuese así bastaría reenumerarlos) y el otro es a_1, a_2, \dots, a_n , de forma que dos productos i y j están invertidos si $i < j$ pero $a_i > a_j$. De esta forma nuestra representación del problema será un vector de enteros v de tamaño n , de forma que $v[i] = a_i$, $i = 1, \dots, n$.

El objetivo es diseñar, analizar la eficiencia e implementar un algoritmo “divide y vencerás” para medir la similitud entre dos rankings. Compararlo con el algoritmo de “fuerza bruta” obvio. Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

3. Eliminar elementos repetidos

Dado un vector de n elementos, de los cuales algunos pueden estar duplicados, el problema es obtener otro vector donde todos los elementos duplicados hayan sido eliminados.

Diseñar, analizar la eficiencia e implementar un algoritmo sencillo para esta tarea, y luego hacer lo mismo con un algoritmo más eficiente, basado en “divide y vencerás”, de orden $O(n \log n)$. Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

4. El elemento en su posición

Dado un vector ordenado (de forma no decreciente) de números enteros v , todos distintos, el objetivo es determinar si existe un índice i tal que $v[i] = i$ y encontrarlo en ese caso. Diseñar e implementar un algoritmo “divide y vencerás” que permita resolver el problema. ¿Cuál es la complejidad de ese algoritmo y la del algoritmo “obvio” para realizar esta tarea? Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

Supóngase ahora que los enteros no tienen por qué ser todos distintos (pueden repetirse). Determinar si el algoritmo anterior sigue siendo válido, y en caso negativo proponer uno que sí lo sea. ¿Segue siendo preferible al algoritmo obvio?

5. Mezclando k vectores ordenados

Se tienen k vectores ordenados (de menor a mayor), cada uno con n elementos, y queremos combinarlos en un único vector ordenado (con kn elementos). Una posible alternativa consiste en, utilizando un algoritmo clásico, mezclar los dos primeros vectores, posteriormente mezclar el resultado con el tercero, y así sucesivamente.

- ¿Cuál sería el tiempo de ejecución de este algoritmo?
- Diseñe, analice la eficiencia e implemente un algoritmo de mezcla más eficiente, basado en “divide y vencerás”.
- Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

6. Serie unimodal de números

Sea un vector v de números de tamaño n , todos distintos, de forma que existe un índice p (que no es ni el primero ni el último) tal que a la izquierda de p los números están ordenados de forma creciente y a la derecha de p están ordenados de forma decreciente; es decir $\forall i, j \leq p, i < j \Rightarrow v[i] < v[j]$ y $\forall i, j \geq p, i < j \Rightarrow v[i] > v[j]$ (de forma que el máximo se encuentra en la posición p). Diseñe un algoritmo “divide y vencerás” que permita determinar p . ¿Cuál es la complejidad del algoritmo? Compárelo con el algoritmo “obvio” para realizar esta tarea. Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

NOTA: Para la realización de los experimentos con los distintos algoritmos se proporcionarán generadores de datos de entrada para cada problema.