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 2018-2019

Grado en Informática

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.

1. Suma de dos elementos

Dado un vector de n enteros, el problema consiste en determinar si existen en el vector dos números cuya suma sea igual a X, para un X dado, y en ese caso identificar esos números.

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.

2. Inversión óptima

Se dispone de una secuencia de n datos correspondientes al valor $p[i] > 0$ de las acciones de una cierta empresa a lo largo de diferentes días, $i = 1, 2, \ldots, n$. Se desea analizar esa secuencia para determinar los dos días c y v (con $c < v$) en que, si hubiésemos comprado acciones el día c y las hubiésemos vendido el día v, el beneficio habría sido el máximo posible (pudiendo ser negativo dicho beneficio). Por ejemplo, si $n = 3$ y $p = [9, 1, 6]$, entonces la decisión óptima sería comprar el día $c = 2$ y vender el día $v = 3$, con un beneficio de 5 unidades. Si $p = [9, 6, 1]$ entonces la "mejor" decisión es comprar el día $c = 1$ y vender el día $v = 2$ con un "beneficio" máximo de -3 unidades.

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.

3. 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 $1, \ldots, n$, y que uno de los rankings siempre es $1, \ldots, n$ (si no fuese así bastaría reenumerarlos) y el otro es $a_1, a_2, \ldots, a_n$, 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 $v[i] = a_i$, $i = 1, \ldots, n$.

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 obvio. Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

4. Sumas parciales de elementos consecutivos

Dados n enteros cualesquiera $a_1, a_2, \ldots, a_n$, se necesita calcular el valor de la expresión:

$$max_{1 \leq i \leq j \leq n} \sum_{k=i}^{j} a_k$$

que calcula el máximo de las sumas parciales de elementos consecutivos. Por ejemplo, dados 6 números enteros (-2, 11, -4, 13, -5, -2), la solución al problema es 20 (suma de a_2 hasta a_4).

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.

5. 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.