Memoria de divide y vencerás

Rubén Morales Pérez Francisco Javier Morales Piqueras
Bruno Santindrian Manzanedo Ignacio de Loyola Barragan Lozano
Francisco Leopoldo Gallego Salido

10 de abril de 2016

Índice

1. Introducción

Divide y vencerás es una técnica algorítmica que consiste en resolver un problema diviéndolo en problemas más pequeños y combinando las soluciones. El proceso de división continúa hasta que los subproblemas llegan a ser lo suficientemente sencillos como para una resolución directa.
El hecho de que el tamaño de los subproblemas sea estrictamente menor que el tamaño original del problema nos garantiza la convergencia hacia los casos elementales. El coste computacional se determina resolviendo relaciones de recurrencia.

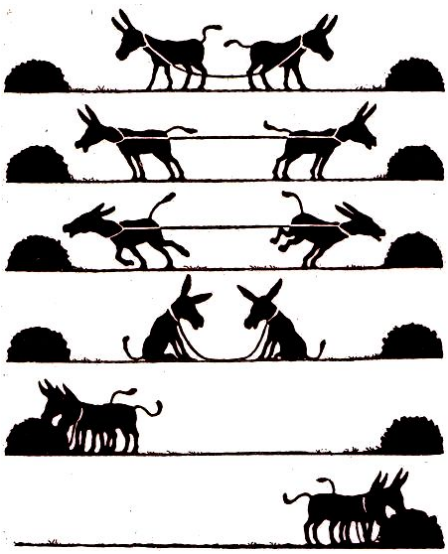

Figura 1: Subproblemas

Una forma de dirigir las cartas siguiendo este método (creada por Donald Knuth, autor de The Art of Computer Programming) es separarlas en bolsas diferentes en función de su área geográfica y cada una a su vez se ordena en lotes para subregiones más pequeñas.
Un ejemplo anterior a Cristo donde se usa ¨divide y vencerás¨ con un único subproblema es el algoritmo de Euclides para computar el máximo común divisor de dos números.

$$
\begin{aligned}
11312 &= 500 \cdot 22 + 312, \ldots [1] \\
500 &= 312 \cdot 1 + 188, \ldots\ldots [2] \\
312 &= 188 \cdot 1 + 124, \ldots\ldots [3] \\
188 &= 124 \cdot 1 + 64, \ldots\ldots\ldots [4] \\
124 &= 64 \cdot 1 + 60, \ldots\ldots\ldots [5] \\
64 &= 60 \cdot 1 + 4 \ldots\ldots\ldots\ldots [6]
\end{aligned}
$$

Figura 2: Euclides

2. 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.

2.1. Tiempo de ejecución

2.2. Mezcla con divide y vencerás

2.3. Estudio empírico e híbrido

3. Comparación de preferencias (opcional)

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

El objetivo es diseñar, analizar la eficiencia e implementar un algoritmo d̈ivide y vencerásp̈ara medir la similitud entre dos rankings. Compararlo con el algoritmo de f̈uerza brutaöbvio. Realizar también un estudio empírico e híbrido de la eficiencia de ambos algoritmos.

3.1. Fuerza bruta

3.2. Divide y vencerás

4. Bibliografia