

ALGORITMOS Y ESTRUCTURAS DE DATOS II

GRADO INGENIERÍA INFORMÁTICA, CURSO 11/12

PRÁCTICA DE DIVIDE Y VENCERÁS Y ANÁLISIS DE ALGORITMOS

A. Contexto

La multiplicación de matrices clásica es una operación que requiere un gran número de cálculos. Continuamente se buscan mecanismos que reduzcan el esfuerzo que realiza, tanto mediante nuevos métodos como aprovechándose de las características especiales que puedan presentar las matrices objeto de la multiplicación. La técnica de divide y vencerás se puede utilizar en algunos de esos casos.

En esta práctica se trata de aplicar técnicas divide y vencerás en la multiplicación de **matrices triangulares** (los elementos por encima o por debajo de las diagonales valen 0).

Se puede reducir el tiempo de ejecución no realizando las multiplicaciones que involucran a la parte triangular compuesta por ceros. Para esto se diseñan algoritmos especiales, y es posible también utilizar técnicas divide y vencerás, descomponiendo el problema en varios subproblemas, de los que algunos pueden ser omitidos (por ejemplo, cuando toda la submatriz correspondiente valga 0). Esta descomposición divide y vencerás, aunque realiza el mismo número de operaciones que la multiplicación directa específica para el tipo de matriz con que se esté trabajando, puede producir una reducción en el tiempo de ejecución debido a una mejora de los accesos a memoria.

Otra posibilidad de utilizar divide y vencerás en la multiplicación de matrices es el algoritmo de Strassen, con el que se reduce la complejidad del algoritmo de multiplicación, pero las operaciones adicionales de menor coste y la gestión de más memoria producen una sobrecarga importante, lo que hace que normalmente solo se obtienen mejores resultados con el algoritmo de Strassen que con el tradicional de tres bucles para matrices muy grandes.

B. Enunciado de la actividad

Esta actividad se hará en grupos de dos alumnos (en casos excepcionales en grupos de un alumno, para lo que habrá que pedir permiso al profesor). Cada grupo tiene asignado un problema distinto, según los identificadores en el juez on-line de la asignatura. Sea X la suma de los números de los identificadores del grupo (quitando la letra inicial), el número del problema asignado es:

$$(X \text{ módulo } 6) + 611$$

Se ofrece a los alumnos una solución base del problema (el programa 600.cpp), usando un método directo y un algoritmo basado en divide y vencerás sencillo (no Strassen). Ambas versiones son implementaciones no optimizadas para el caso de las matrices triangulares.

El trabajo a realizar por los alumnos consistirá en lo siguiente:

1. Optimizar el método directo, evitando los productos por cero. Escribir una implementación que logre la aceptación en el juez on-line de la asignatura.
2. Optimizar el método recursivo de divide y vencerás, evitando los productos de submatrices que valgan cero. Escribir una implementación que logre la aceptación en el juez on-line de la asignatura.
3. Realizar una implementación de la multiplicación de Strassen que no explote la estructura de ceros para matrices de tamaño potencia de dos.
4. Realizar una implementación de la multiplicación de Strassen que explote la estructura de ceros para matrices triangulares de tamaño potencia de dos.
5. Analizar de forma teórica el tiempo de ejecución de las cuatro versiones implementadas. Comparar los órdenes de complejidad.
6. Validar los cuatro algoritmos implementados. La validación puede ser en algunos casos a través del juez online, pero también se puede implementar un programa en el que se generen aleatoriamente matrices, se realice la multiplicación de las distintas formas implementadas y se compare automáticamente si los resultados son correctos. Habrá que justificar que la validación realizada es correcta.
7. Realizar un estudio experimental de los tiempos de ejecución de los cuatro algoritmos para matrices de distintos tamaños, hasta un máximo de 2048. Comparar los resultados experimentales con los distintos algoritmos.
8. Contrastar los resultados teóricos con los experimentales.
9. Realizar alguna modificación de alguno de los algoritmos directos para mejorar el acceso a la memoria. Utilizar para esto las ideas vistas en la sesión de Prácticas de Influencia del Uso de Memoria en las Prestaciones.

Es requisito indispensable programar los cuatro algoritmos propuestos en C/C++, y conseguir la aceptación de los cuatro en el juez on-line de la asignatura: <http://dis.um.es/~mooshak>, **concurso “AED 11/12: Divide y vencerás”**. En otro caso, el problema no será tenido en cuenta. Observar, no obstante, que la aceptación en el juez on-line no es requisito suficiente para aprobar la actividad.

C. Memoria de la actividad

La memoria entregada deberá contener los siguientes apartados.

C.1. Portada

Nombre del alumno o alumnos, e-mail y número de login en Mooshak de la cuenta que han usado para realizar los envíos.

C.2. Lista de problemas resueltos

Lista de los envíos realizados al juez on-line, indicando a que implementación corresponde cada uno, y la causa de los fallos que se haya producido.

C.3. Resolución de problemas

Para el problema asignado al grupo, se deberán incluir los siguientes apartados:

1. (hasta **3** puntos) Códigos de los cuatro algoritmos implementados y explicación de cada algoritmo, justificando las decisiones de diseño. Los programas deben ir documentados, con explicación de qué es cada variable y qué realiza cada función.
2. (hasta **1** punto) Validación: habrá que justificar la forma en que se validan los resultados para estar seguros de que se obtienen los resultados correctos. Además de ser aceptados por Mooshak se puede diseñar algún procedimiento de validación.
3. (hasta **1.5** puntos) Estudio teórico del tiempo de ejecución de cada uno de los algoritmos, y comparación de los tiempos de ejecución, determinando para qué tamaños de problema es preferible cada algoritmo, y los tamaños óptimos del caso base en los algoritmos divide y vencerás.
4. (hasta **1** puntos) Estudio teórico de la ocupación de memoria en los cuatro algoritmos, y comparación de las necesidades de memoria.
5. (hasta **1.5** puntos) Estudio experimental del tiempo de ejecución para distintos tamaños de problema y del caso base, y comparación de los tiempos de ejecución, determinando para qué tamaños de problema es preferible cada algoritmo, y los tamaños óptimos del caso base en los algoritmos divide y vencerás.
6. (hasta **1** punto) Contraste de estudio teórico y experimental para los distintos algoritmos, buscando justificación a las discrepancias entre los dos estudios.
7. (hasta **1** punto) Análisis de la reducción del tiempo de ejecución por modificaciones en el acceso a memoria, para alguno de los algoritmos programados. Incluye implementación de versión con mejora de acceso a memoria, estudio experimental y comparación de los resultados experimentales con los obtenidos con el algoritmo original.

C.4. Conclusiones

Incluir las conclusiones y valoraciones personales de la actividad, y la estimación del tiempo total que se ha tardado en completarla.

D. Evaluación de la actividad

La documentación generada se enviará por correo en pdf al profesor de prácticas, a su correo personal o a través del aula virtual. El concurso “AED 11/12: Divide y vencerás” del juez on-line **se cerrará el viernes 20 de abril de 2012 a las 22:00**. La fecha tope de entrega de la documentación será a lo largo de ese mismo día.

El profesor realizará una entrevista individual con cada uno de los alumnos, aunque puede decidir no realizar la entrevista con alguno de ellos si ha seguido su trabajo en las sesiones de prácticas. Las entrevistas se realizarán en las sesiones de prácticas.

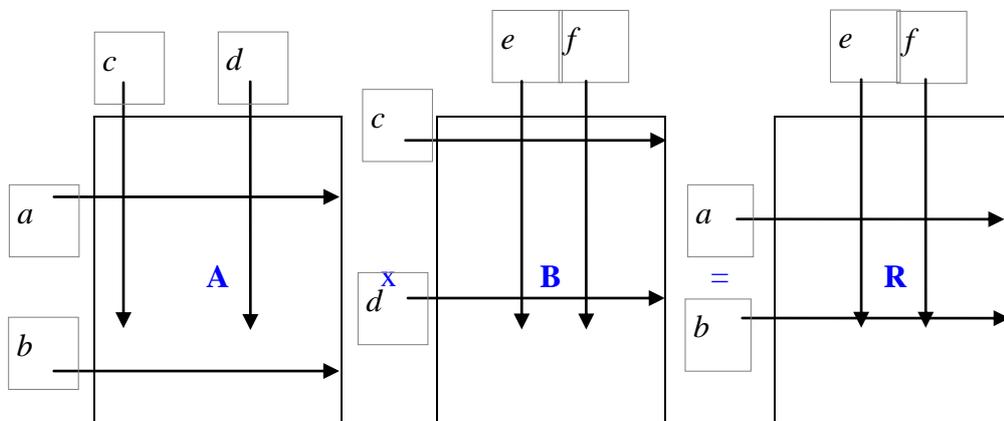
La puntuación de los distintos apartados son las que se muestran en la sección C.1 de este documento. Es condición necesaria para superar la práctica que los cuatro métodos implementados en la sección C.1.1 sean admitidos en mooshak. Además, habrá que obtener al menos un 5 en la puntuación total de la práctica. La participación en las dos sesiones de laboratorio iniciales (análisis e influencia del uso de memoria en las prestaciones) pueden subir hasta un 0.3 máximo por cada una de ellas, siendo la puntuación máxima final de la práctica de 10.

E. Material disponible

Para realizar esta actividad, se aconseja que los alumnos utilicen los programas ofrecidos por los profesores:

- **Implementación básica (no optimizada) por el método directo y por divide y vencerás (en 600.cpp)**. Esta versión resuelve el problema de multiplicar matrices completas, por lo que no está optimizada para los casos de matrices triangulares.

Ojo: en este programa todas las funciones para multiplicar matrices trabajan con las matrices globales **A**, **B** y **R**, donde **A** y **B** son las matrices de entrada y **R** es el resultado. Las funciones reciben seis parámetros (a , b , c , d , e , f) que indican las posiciones que hay que multiplicar en estas matrices, según el siguiente esquema:



Es decir, se multiplican las filas ($a-b$) y columnas ($c-d$) de la matriz **A**, por las filas ($c-d$) y columnas ($e-f$) de **B**, almacenando el resultado en **R**, en las filas ($a-b$) y

columnas ($e-f$). De esta forma, nos evitamos tener que reservar memoria para los distintos subproblemas: siempre se trabaja con matrices globales.

- **Generador de casos de prueba (genera600.cpp)**. Este programa sirve para generar distintos casos de prueba (matrices), con tamaños y tipos especificados por el usuario, según el formato de los problemas del juez on-line.
- Las implementaciones de la multiplicación de matrices utilizadas en las sesiones de prácticas.

AVISO IMPORTANTE

Las entregas de todos los alumnos, en todas las convocatorias y titulaciones, serán sometidas a un sistema automático de **detección de plagios**. Copiar cualquier ejercicio de otro grupo supondrá el suspenso fulminante, ojo, no sólo de la práctica sino de **toda la asignatura**, para todos los alumnos implicados.