Informe

Análisis y diseño de Algoritmos

1er entrega

Alumno: Attilio, Nicolas

Comision : 1

Ayudante: Lautaro, Valenzuela

En este informe se detalla la especificación y la implementación del algoritmo de Strassen, el cual calcula la multiplicación de dos matrices.

Además evaluaremos su complejidad temporal y haremos un análisis con respecto a otros algoritmos para ver su efectividad o no, en cuanto a los mecanismos de los tiempos y de espacios

Algoritmo de Strassen

Sean A y B dos matrices de dimensiones 2kx2k particionadas en 4 matrices de 2k-1x2k-1:



Entonces hacemos las siguientes operaciones:

I=(A11+A22)(B11+B22)

II=(A21+A22)B11

III=A11(B11-B12)

IV=A22(B21-B22)

V=(A11+A12)B22

VI=(A21-A11)(B11+B12)

VII=(A22-A12)(B21+B22)

C11=I+IV-V+VII

C12=III+V

C21=II+IV

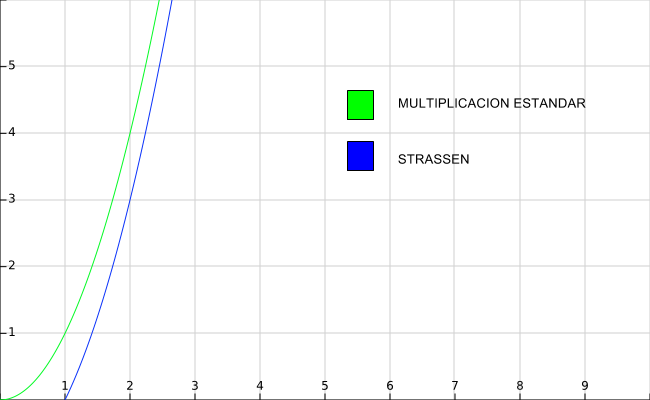
C22=I+III-II+IV

Así queda formado la matriz C con 7 multiplicaciones de orden 2k-1x2k-1

Analisis

El cálculo de una multiplicación de dos matrices ralas lleva 8 operaciones por lo tanto la complejidad temporal es de T(n)O(n3)

En cambio el algoritmo de Strassen lleva 7 operaciones para llegar al mismo resultado, por lo tanto hay una leve mejora en la complejidad temporal que da como resultado T(n)O(nlog 7)=(n2.8074)



Como vemos en el gráfico en función del tiempo, hay una leve mejora del algoritmo de Strassen con respecto a la multiplicación estándar.

En cuanto a nuestro codigo si se lo aplicamos a matrices ralas, tendríamos que completar con números 0(cero) para que sean matrices cuadradas, lo cual conlleva una pérdida de espacio pero no de tiempo, ya que el peor caso ya está dado en la antigua condición.

Las dificultades que nos enfrenta agregar este codigo seria la perdida de referencia sobre algunos valores, ya que se modifican con los agregados anteriormente, y así mismo la asignación a la matriz resultante.

Una alternativa seria que si se encuentra con un valor 0 podría ser obviado, lo cual es una solución rápida pero no efectiva ya que la matriz tendría valores mayores a 0.

Otra modificación para considerar que las matrices no fueran de tamaño de potencia de 2 sería asignar todos los datos a un arreglo dinámico de tamaño de la potencia junto con su exponente y trabajarlos de forma lineal. Pero sería muy trabajoso a la hora de hacer los algoritmos necesarios para resolverlo.

Especificación informal de los TDA

La estructura con la que se trabajó fue de una matriz dinámica a la cual en la construcción se le arma agregando a todos sus campos el valor de 0.

Luego en el getvalue se devuelve el valor en la posición deseada y el set value se asigna un valor en otra posición deseada.

La suma y la resta se trabaja con la matriz seleccionada y se hace las operaciones sobre las matrices pasadas por parámetro.

La multiplicación directa toma la matriz C y la iguala a la multiplicación de las matrices A y B.

En la recursión se dividió la matriz en 4 cuadrantes por la longitud / 2, que seria lo mismo decir 2^k-1 y cuando encuentro la longitud mínima hago las 7 operaciones.

La complejidad temporal está evaluada por el valor de la potencia, el mejor caso sería T(n)O(n3) y el peor caso sería T(n)O(2n).

Esta estructura perjudica el costo de la solución ya que trabajándolo como un arreglo se puede bajar la complejidad temporal, en el cual no tiene un doble recorrido sobre el tipo de dato.

Resumen

En este trabajo pudimos crear, modificar y desarrollar matrices dinámicas y otros tipos de datos para resolver esta tarea.

Además, se intentó implementar que el código fuera con la menor complejidad posible para que este sea eficiente y sencillo a la hora de visualizarlo y de ejecutarlo.