

Aplicando la técnica de Divide-and-Conquer

1. Introducción

El primer objetivo de este laboratorio es agregar a la librería de ordenamiento `Sortlib.kt`, una versión del algoritmo Mergesort. El segundo objetivo es implementar el algoritmo de Strassen para la multiplicación de matrices.

2. Actividades a realizar

En esta sección se presentan las actividades que debe realizar para cumplir los objetivos del laboratorio.

2.1. Algoritmo Mergesort

Se quiere que implemente la versión de Mergesort presentada en la página 229 de [1]. En este algoritmo se tiene que cuando el número de elementos a ordenar es menor o igual a un valor X , entonces se ordena el arreglo con *Insertion-Sort*. También se quiere determinar experimentalmente el mejor valor para X . Para lograr estas metas, en primer lugar debe agregar a la librería `Sortlib.kt` un procedimiento llamado `mergesortInsertion` que es la implementación del algoritmo de Mergesort indicado anteriormente. En segundo lugar, debe realizar un estudio experimental para determinar el tamaño máximo de X que debe tener el arreglo para aplicar el algoritmo de *Insertion-Sort*, de manera que el tiempo de ejecución de Mergesort, sea el mejor posible en la práctica.

El estudio experimental que debe realizar es como sigue. Debe crear un procedimiento llamado `mergesortInsertionPrueba`, que es una modificación `mergesortInsertion`, que recibe el arreglo a ordenar, y un número n , que es el tamaño que debe tener el arreglo para aplicar *Insertion-Sort*. Para el estudio experimental, se debe crear un programa de pruebas, que debe estar contenido en un archivo llamado `PruebaMergesort.kt`. El procedimiento `mergesortInsertionPrueba`, debe estar contenido en este archivo. El programa crea un arreglo de un millón de números enteros generados aleatoriamente, en un rango de $[0, 900,000]$. El mismo arreglo se ordena con `mergesortInsertionPrueba` aplicando *Insertion-Sort* a arreglos de tamaños 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100. Para cada prueba se debe tomar el tiempo, en segundos, que toma `mergesortInsertionPrueba`.

Una vez obtenido los resultados de ejecutar `PruebaMergesort.kt`, debe realizar un reporte, de un hoja y en formato PDF, en el que se muestre los resultados obtenidos para los 10 tamaños de arreglo, para los se aplicó *Insertion-Sort*. Los resultados deben estar contenidos en una tabla que muestre el tamaño del arreglo y el tiempo del procedimiento `mergesortInsertionPrueba`. En el reporte también debe indicar los datos del sistema en donde hicieron las pruebas: el sistema de operación, el modelo CPU y la cantidad de RAM del computador.

Finalmente, el mejor valor obtenido en el estudio experimental, es el que debe ser colocado en el procedimiento `mergesortInsertion`, que debe estar incluido en la librería `Sortlib.kt`.

2.2. Multiplicación de matrices

La segunda actividad consiste en la implementación de algoritmos de multiplicación de matrices de números enteros. El archivo en donde van a estar los algoritmos se debe llamar `MultiplicacionMatrices.kt`. Ambos algoritmos de multiplicación de matrices reciben como entrada dos matrices cuadradas de $N \times N$ elementos, y la dimensión N de las matrices. El primer algoritmo corresponde al algoritmo básico de multiplicación de matrices, el cual debe ser implementado en una función llamada `multiplicacionSimpleDeMatrices`. El segundo término debe implementar el algoritmo de Strassen, el cual debe estar contenido en una función llamada `multiplicacionStrassen`. La versión del algoritmo de Strassen que debe implementar, es la dada en [2].

3. Condiciones de entrega

La versión final del código del laboratorio y la declaración de autenticidad firmada, deben estar contenidas en un archivo comprimido, con formato *tar.xz*, llamado *LabSem2_X_Y.tar.xz*, donde X y Y , son los números de carné de los estudiantes. La entrega del archivo *LabSem2_X_Y.tar.xz*, debe hacerse por la plataforma Classroom, antes de las 11:50 pm del día domingo 21 de mayo de 2023.

Referencias

- [1] BRASSARD, G., AND BRATLEY, P. *Fundamentals of Algorithmics*. Prentice Hall, 1996.
- [2] T. CORMEN, C. LEIRSESON, R. R., AND STEIN, C. *Introduction to Algorithms*. McGraw Hill, 2022.