

# Tarea 1 - Ordenamiento

CC4102/CC53A - Diseño y Análisis de Algoritmos  
Profesor: Pablo Barceló    Auxiliar: Miguel Romero

Fecha de Entrega: 05 de Mayo del 2014

## 1 Introducción

El objetivo de esta tarea es implementar y evaluar en la práctica distintos algoritmos de ordenamiento en memoria principal. Consideraremos cuatro algoritmos: *Bubble Sort*, *Insertion Sort*, *Merge Sort* y *Quick Sort*. Para la evaluación se considerarán dos tipos de instancias: instancias *aleatorias* e instancias *casi ordenadas*. Se espera que el alumno implemente individualmente los algoritmos y entregue un informe que indique claramente los siguientes puntos:

1. Las *hipótesis* escogidas **antes** de realizar los experimentos.
2. El *diseño experimental*, incluyendo los detalles de la implementación de los algoritmos, la generación de las instancias y las medidas de rendimiento utilizadas.
3. La *presentación de los resultados* en forma de una descripción textual, tablas y/o gráficos.
4. El *análisis e interpretación* de los resultados.

## 2 Algoritmos

Todos los algoritmos reciben un arreglo  $A[1, n]$  de números naturales, y retornan otro arreglo  $B[1, n]$ , que corresponde al arreglo  $A$  ordenado de menor a mayor. Sólo consideraremos algoritmos basados en comparaciones. A continuación se da una breve descripción de cada uno de ellos:

1. **Bubble Sort:** Se recorre  $A[1, n]$  de izquierda a derecha, corrigiendo pares de elementos consecutivos que estén mal ordenados (es decir, intercambiamos  $A[i]$  y  $A[i + 1]$ , si  $A[i] > A[i + 1]$ ). Repetimos esto hasta que no haya nada que corregir.
2. **Insertion Sort:** Se recorre  $A[1, n]$  de izquierda a derecha y mantenemos el siguiente invariante: En la  $i$ -ésima iteración, el arreglo  $A[1, i - 1]$  está ordenado e insertamos  $A[i]$  en la posición que le corresponde dentro de  $A[1, i]$ .
3. **Merge Sort:** Recursivamente ordenamos  $A[1, n/2]$  y  $A[n/2, n]$  y luego mezclamos las dos soluciones con una pasada de izquierda a derecha.
4. **Quick Sort:** Escogemos  $A[n]$  como pivote, y particionamos  $A[1, n]$  en  $A[1, i - 1]$  (los menores que  $A[n]$ ), y  $A[i + 1, n]$  (los mayores que  $A[n]$ ). Ordenamos recursivamente  $A[1, i - 1]$  y  $A[i + 1, n]$ .

El informe debe indicar el detalle de la implementación de cada algoritmo.

### 3 Experimentos

Consideraremos instancias de tamaño  $n \in \{2^{10}, 2^{11}, \dots, 2^{20}\}$ . Las medidas de rendimiento que usaremos son:

- Número esperado de comparaciones.
- Tiempo esperado de ejecución.

Para generar instancias aleatorias, puede utilizar el siguiente algoritmo:

**GeneradorAleatorio( $n$ ):**

1.  $A = [1, 2, \dots, n]$
2. for  $i = 1$  to  $n$ :
3.     swap  $A[i]$  with  $A[\text{Random}(i, n)]$

La función  $\text{Random}(i, n)$  entrega un natural en el rango  $[i, n]$ , cada uno con igual probabilidad. Se puede demostrar que **GeneradorAleatorio( $n$ )** genera cada permutación de  $[1, 2, \dots, n]$  con igual probabilidad (Cormen, sección 5.3).

Para cada  $n \in \{2^{10}, 2^{11}, \dots, 2^{20}\}$  debe repetir sus mediciones hasta que el margen de error sea menor que el 5% (es decir, hasta que la desviación estándar sea a lo más 0.05 veces la media). Si esto no es posible, o exige demasiadas repeticiones, indique su margen de error alcanzado.

Finalmente, queremos evaluar los algoritmos bajo instancias casi ordenadas. Una instancia está casi ordenada si gran parte de sus elementos ya están en su posición correcta. Usted deberá diseñar los experimentos en este caso (la generación de instancias y los parámetros utilizados).

### 4 Entrega de la Tarea

- La tarea puede realizarse en grupos de a lo más 2 personas.
- Para la implementación puede utilizar **C**, **C++**, **Java** o **Python**. Para el informe se recomienda utilizar **L<sup>A</sup>T<sub>E</sub>X**.
- Escriba un informe claro y conciso. Las ponderaciones del informe y la implementación en su nota final son las mismas.
- La entrega será a través de U-Cursos y deberá incluir el informe junto con el código fuente de la implementación (y todas las indicaciones necesarias para su ejecución).