



Análisis y Diseño de Algoritmos

Grado en Ingeniería Informática

Práctica 1 - curso 2023/24

Introducción

El objetivo de la práctica es comparar tres algoritmos de ordenación. Para ello es preciso obtener una estimación, basada en datos experimentales, de la eficiencia **en el caso promedio** de los algoritmos. Una vez hecha la estimación para los tres algoritmos, hay que determinar, si fuera posible, la curva que más se aproxima a los datos obtenidos. Por último, hay que determinar cuál de los algoritmos es más eficiente.

Algoritmo 1

```
void ordenal(int v[], int tam) {
    /* v con índices de 0 a tam-1 */
    int h, r, i, j, w;
    r = tam-1;
    h = 1;
    while (h <= r/9) {
        h = 3*h+1;
    }
    while (h > 0) {
        for (i = h; i <= r; i++) {
            j = i;
            w = v[i];
            while ((j >= h) && (w < v[j-h])) {
                v[j] = v[j-h];
                j = j - h;
            }
            v[j] = w;
        }
        h = h / 3;
    }
}
```



Algoritmo 2

```
void ordena2(int v[], int tam) {
    /* v con índices de 0 a tam-1 */
    boolean swapped = true;
    int start = 0;
    int end = tam;

    while (swapped == true) {
        swapped = false;
        for (int i = start; i < end - 1; ++i) {
            if (v[i] > v[i + 1]) {
                int temp = v[i];
                v[i] = v[i + 1];
                v[i + 1] = temp;
                swapped = true;
            }
        }
        if (swapped == false) {
            break;
        }
        swapped = false;
        end = end - 1;
        for (int i = end - 1; i >= start; i--) {
            if (v[i] > v[i + 1]) {
                int temp = v[i];
                v[i] = v[i + 1];
                v[i + 1] = temp;
                swapped = true;
            }
        }
        start = start + 1;
    }
}
```



Algoritmo 3

```
void ordena3(int[] v, int tam) {
    /* v con índices de 0 a tam-1 */
    int w[] = new int[tam];
    ordena3rec(v, w, 0, tam - 1);
}

void ordena3rec(int v[], int w[], int l, int r) {
    if (l < r) {
        int m = (l + r) / 2;
        ordena3rec(v, w, l, m);
        ordena3rec(v, w, m + 1, r);
        int ia = l;
        int ib = m + 1;
        int ic = l;
        while ((ia <= m) && (ib <= r)) {
            if (v[ia] < v[ib]) {
                w[ic] = v[ia];
                ia++;
                ic++;
            } else {
                w[ic] = v[ib];
                ib++;
                ic++;
            }
        }
        while (ia <= m) {
            w[ic] = v[ia];
            ia++;
            ic++;
        }
        while (ib <= r) {
            w[ic] = v[ib];
            ib++;
            ic++;
        }
        for (int i = l; i <= r; i++) {
            v[i] = w[i];
        }
    }
}
```



Desarrollo de la práctica

La realización de la práctica constará de dos partes. En la primera se contarán las operaciones elementales y en la segunda se medirá el tiempo empleado. En ambos casos, hay que construir un programa para obtener los datos:

- Se crearán subprogramas que implementen cada uno de los algoritmos.
- Se crearán y asignarán aleatoriamente vectores de distintos tamaños. Por ejemplo 10.000, 20.000, ..., 100.000. Para cada tamaño se llamará un cierto número de veces (por ejemplo 20) al subprograma. Debe tenerse en cuenta que antes de cada llamada al subprograma se deberá volver a asignar valores aleatorios al vector.
- El programa nos servirá para obtener los datos necesarios para analizar los algoritmos. Proporcionará como salida las tablas que indiquen:
 - El número de comparaciones y asignaciones promedio según el tamaño del vector. Se considerarán solamente las operaciones (comparaciones y asignaciones) **en las que intervienen elementos del vector** (no las comparaciones y asignaciones entre índices, por ejemplo).
 - El tiempo empleado según el tamaño del vector.

Con los datos anteriores se construirán tres gráficas, una para el número de comparaciones, otra para el número de asignaciones y otra para el tiempo empleado. A partir de ellas se intentará obtener una fórmula que las describa en función del tamaño del vector. Por ejemplo, se pueden ajustar las estimaciones a una curva polinómica, logarítmica, etc.

Contar operaciones elementales

Para contar las operaciones elementales se añadirán las instrucciones necesarias en el programa para contar el número promedio de comparaciones y el número promedio de asignaciones. Se considerarán solamente las comparaciones y asignaciones **en las que intervengan elementos del vector** (no las comparaciones y asignaciones entre índices, por ejemplo).

Medir el tiempo empleado

La medición del tiempo se puede realizar de varias maneras. Una de las más sencillas es emplear las clases *Instant* y *Duration* de Java. En ocasiones la medición del tiempo no es muy precisa, sobre todo cuando el tamaño del vector es pequeño.

```
import java.time.Duration;
import java.time.Instant;
...
Instant start = Instant.now();
// Código que queremos medir
Instant finish = Instant.now();
long timeElapsed = Duration.between(start, finish).toNanos();
```

Análisis de datos

Para analizar los datos se sugiere utilizar programas que permitan representar gráficamente los datos obtenidos, por ejemplo LibreOffice Calc, Microsoft Excel o cualquier herramienta conocida que sea útil para este tipo de tareas.



Normas de entrega

La práctica se realizará en **grupos de 2 personas**.

Se deberá entregar el código fuente de los programas que se han creado para la realización de la práctica (se recomienda el uso del lenguaje Java aunque alternativamente se permite realizar en cualquier lenguaje de programación). Dichos programas deben compilar y ejecutar perfectamente en las máquinas de los laboratorios. Estos programas deben generar como salida un fichero con los datos promedio: tamaño del vector, tiempo promedio, comparaciones promedio y asignaciones promedio.

Además se entregará un documento en **formato PDF** donde se indique lo siguiente:

- Nombre de los alumnos.
- **Muy importante:** hay que estimar como se ha repartido el trabajo entre los miembros del equipo. Para ello, **hay que indicar el porcentaje de esta entrega que se estima ha desarrollado cada miembro del equipo.**
- Una breve descripción de cómo se ha realizado la práctica (rango de tamaños y repetición de medidas, programas utilizados para el análisis, etc.).
- Las tablas y los gráficos con los resultados obtenidos en las medidas experimentales de los algoritmos.
- La fórmula obtenida para cada uno de los algoritmos.
- **Una explicación RAZONADA de qué algoritmo es mejor.**

La fecha límite de entrega es el **27 de octubre a las 14:00**. Los ficheros de la práctica se empaquetarán en un fichero zip cuyo nombre será `p1_grupoX.zip`, siendo X el número del grupo de laboratorio. El fichero zip se entregará a través del Campus Virtual (<https://campusvirtual.uva.es/>).

La defensa de la práctica es obligatoria y se realizará en la sesión de laboratorio de la semana siguiente a la entrega:

- Grupo L1: miércoles 8 de noviembre a las 10:00.
- Grupo L2: jueves 2 de noviembre a las 12:00.