

# Proyecto 1: Trazas de Ejecución

**Martes 3 de Septiembre**

## I. DESCRIPCIÓN GENERAL

En este proyecto, se mostrará la traza de ejecución de varios algoritmos de ordenamiento. Se debe generar un documento  $\text{\LaTeX}$  el cual será compilado internamente para generar un PDF que a su vez será desplegado en modo presentación. Toda la programación debe hacerse en Lenguaje C sobre Linux.

## II. PRELIMINAR

Aunque sigue un enfoque diferente a este proyecto, deben leer el artículo “*An Analysis of Sorts*” de Jonathan Amsterdam y entenderlo. Viene adjunto a esta especificación.

## III. ALGORITMOS

Su código debe incluir 6 de los siguientes algoritmos:

1. **Insertion Sort**
2. **Exchange Sort**
3. **Merge Sort**
4. **Quick Sort**
5. **Bubble Sort**
6. **Shell Sort**
7. **Gnome Sort**
8. **Cocktail Sort**
9. **Stooge Sort**

Todos programados por ustedes mismos.

**Trabajo extra opcional 1:** Programen **todos** los algoritmos. No hay crédito por 7 u 8 algoritmos.

## IV. DATOS

La entrada siempre será una hilera de 16 caracteres, todos ellos letras del alfabeto en mayúscula que se leerá del *standard input*. Independiente de la entrada del usuario sólo se toman los primeros 16 caracteres. Se hará una validación rápida de que la entrada cumpla los requisitos, si no fuera así se rechaza la hilera y se pide de nuevo.

Después se ejecutan TODOS los algoritmos de ordenamiento que su proyecto incluya (6 ó 9) para ordenar cada vez las 16 letras mayúsculas ingresadas. Para cada uno de ellos se genera una “traza de ejecución” con el mismo formato y estilo que el ejemplo de la Tabla 1 para el **Merge Sort**. Los caracteres que cambian de posición con respecto a la línea previa o que aunque no cambien son importantes en esa iteración del algoritmo particular se presentan en **negrita** y subrayados. Cada traza vendrá claramente identificada con el nombre del algoritmo.

	-	-	-	-	-	-	-	-	-	-	1	1	1	1	1	1
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
	L	D	Q	F	D	R	N	Q	S	D	W	A	L	O	K	D
1	D	L	Q	F	D	R	N	Q	S	D	W	A	L	O	K	D
2	D	L	F	Q	D	R	N	Q	S	D	W	A	L	O	K	D
3	D	F	L	Q	D	R	N	Q	S	D	W	A	L	O	K	D
4	D	F	L	Q	D	R	N	Q	S	D	W	A	L	O	K	D
5	D	F	L	Q	D	R	N	Q	S	D	W	A	L	O	K	D
6	D	F	L	Q	D	N	Q	R	S	D	W	A	L	O	K	D
7	D	D	F	L	N	Q	Q	R	D	S	W	A	L	O	K	D
8	D	D	F	L	N	Q	Q	R	D	S	W	A	L	O	K	D
9	D	D	F	L	N	Q	Q	R	D	S	A	W	L	O	K	D
10	D	D	F	L	N	Q	Q	R	A	D	S	W	L	O	K	D
11	D	D	F	L	N	Q	Q	R	A	D	S	W	L	O	K	D
12	D	D	F	L	N	Q	Q	R	A	D	S	W	L	O	D	K
13	D	D	F	L	N	Q	Q	R	A	D	S	W	D	K	L	O
14	D	D	F	L	N	Q	Q	R	A	D	D	K	L	O	S	W
15	A	D	D	D	D	F	K	L	L	N	O	Q	Q	R	S	W

Tabla 1: Trazas de Ejecución de **Merge Sort**

## V. SALIDA

Su programa debe crear un archivo fuente de  $\text{\LaTeX}$  que al ser compilado con `pdflatex` genere un PDF con las trazas de todos los algoritmos debidamente etiquetados. A manera de ayuda, el código  $\text{\LaTeX}$  correspondiente a la Tabla 1 se muestra en la **Figura 1**.

El documento generado debe identificar al proyecto, el curso, semestre y los miembros del grupo en la primera página. La compilación se debe realizar de manera interna en su programa<sup>1</sup>, y, de manera similar, el PDF producido debe ser desplegado con `evince` en **modo presentación** invocado internamente por su programa.

Después de corrido su programa, **no** se deben borrar ni el fuente de  $\text{\LaTeX}$  ni el PDF generados.

## VI. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo “*todo está bien pero no pudimos pegarlo*”<sup>2</sup> provocan la cancelación automática de la revisión.
- La presentación debe ser de mucha calidad.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

<sup>1</sup>Investigue el uso de la función `system()`

<sup>2</sup>esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.

