

**ISIS 2112 Diseño de Algoritmos**  
**Semestre 2026-10. Proyecto – PARTE 1**  
**Entrega: lunes, 2 de marzo de 2026 6:00PM (por Bloque Neon)**

## 0 OBJETIVOS

- Diseñar soluciones computacionales para problemas.
- Estimar costos de las soluciones planteadas.
- Implementar soluciones.

Se premiarán las mejores soluciones y se castigarán las peores, en cuanto a optimización y eficiencia en tiempo y espacio.

## 1 CONDICIONES GENERALES

El proyecto se divide en tres partes independientes entre sí. Este documento describe la PARTE I. Cada parte contiene un problema a resolver mediante soluciones implementadas en *Java* o *Python*.

Para cada problema se pide:

- Descripción de la solución.
- Análisis temporal y espacial.
- Una implementación en Java o Python

## 2 DESCRIPCIÓN DEL PROBLEMA

En composición tipográfica, un "río" es una cadena de espacios formada por separaciones entre las palabras que se extiende por varias líneas de texto. Por ejemplo, la figura muestra varios ejemplos de ríos resaltados en rojo (en el texto todas las letras han sido remplazadas por el carácter 'x' para hacer que los ríos sean más visibles).

```
xxxx*xxx xxxxx xxxxxxxxxxxxxxxx*xxxxxxxxxxxxxxxxxxxxxx*x xxxxxxxx
xxx*xxxxxxxx*xx xxxxxxxx x*xxxx*xxxxxxxxxxx xxxx xx*xx xxxxxxxx
xx*xxxxxxxx*xxxxxxxx xxxxxx*xxxxxxxxxxx xxxxxxxxxxx xx*xxxx xxxx
xxxxx xxxxx*xxxxxxxxxxx xxxxxx*x xxxxx xxxxxxxxxxxxxxxxxx xxxxxxxx
xxxxxxxxxx xx*xxxxxxxxxxx xxxxxx*x xxxxxx xxxxxx xxxxxxxxxxxxxx xxxx
xx xxxx xxxxx xxxxxxx xxx xxxxxxxxxxxxxxxxxxxxxx xxxx xxxxxxxxxxxxxx
```

El texto que se considera se entiende escrito en una fuente monoespaciada (todas las letras y espacios tienen igual ancho) y en una columna alineada a la izquierda de cierto tamaño fijo, con exactamente un espacio separando palabras en cada línea (el texto no está alineado a la derecha). Un río se define como una secuencia de espacios en líneas consecutivas en las que la posición de cada espacio en la secuencia difiere a lo sumo en 1, desde la posición del espacio en la línea de arriba. Como excepción, ningún espacio en blanco al final de una línea puede aparecer en un río. Las palabras deben estar lo más ajustadas posible en líneas; ninguna palabra puede dividirse entre líneas. El tamaño de línea usado debe ser al menos tan largo como la palabra más larga en el texto.

### Problema:

Dado un texto, se quiere determinar el ancho de línea que produce el río más largo de espacios para ese texto y, en este caso, el largo de dicho río. Si el río más largo se produce con más de un ancho de línea, se debe calcular el ancho más pequeño que lo consigue.

### Ejemplo:

La siguiente tabla muestra el mismo texto configurado con dos anchos de línea diferentes.

Ancho de línea: 16	Ancho de línea: 25
El*Amazonas es	El Amazonas es el río más
el*río más largo	largo de*Suramérica y el
de*Suramérica y	Magdalena*es el más largo
el*Magdalena es	que fluye*enteramente en
el*más largo que	Colombia
fluye	
enteramente en	
Colombia	
Río más largo: 5	Río más largo: 3

## 3 ENTRADA Y SALIDA DE DATOS

En todas las soluciones que se presenten, la lectura de los datos de entrada se hace por la entrada estándar; así mismo, la escritura de los resultados se hace por la salida estándar.

Puede suponer que ninguna línea de entrada tiene espacios al principio o al final, y que los datos que se listan en cada línea están separados por exactamente un espacio.

A continuación, se establecen parámetros que definen su tamaño y formato de lectura de los datos, tanto de entrada como de salida.

### *Descripción de la entrada*

La primera línea de entrada especifica el número de casos de prueba que contiene el archivo. El programa debe terminar su ejecución, una vez termine de resolver la cantidad de casos de prueba dados por este número.

Cada caso de prueba contiene una sola línea con el texto de tamaño  $n$  que se debe analizar. Cada palabra de cada texto consiste en letras minúsculas y mayúsculas del alfabeto español, separadas por un solo espacio. Ninguna palabra tiene más de 80 caracteres.

Tamaño de entrada:  $1 \leq n \leq 10^5$

### *Descripción de la salida*

Para cada caso de prueba se debe producir una línea de la forma

A R

donde A es el ancho de línea más corto que produce un río de tamaño R, siendo R la longitud máxima de río que se puede encontrar en el texto variando el ancho dentro de los límites posibles.

### *Ejemplo de entrada / salida*

Entrada	Salida
El Amazonas es el río más largo de Suramérica y el Magdalena es	16 5
el más largo que fluye enteramente en Colombia	10 5
Mambrú se fue a la guerra que dolor que dolor que pena	15 6
Carito me habla en ingles qué bonito se le ve Carito me habla	
en ingles qué me dice yo no se	

**Nota:** Se van a diseñar casos de prueba para valores de  $n$  mucho más grandes y dentro de los valores establecidos en el enunciado. Los casos mostrados en este documento son demostrativos de la estructura de entrada/salida esperada.

## 5 ENTREGABLES

El proyecto puede desarrollarse por grupos de hasta dos estudiantes de la misma sección. La entrega se hace por bloque neon (una sola entrega por grupo de trabajo).

El grupo debe entregar, por bloque neon, un archivo de nombre `proyectoDalgoP1.zip`. Este archivo es una carpeta de nombre `proyectoDalgoP1`, comprimida en formato `.zip`, dentro de la cual hay archivos fuente de soluciones propuestas y archivos que documentan cada una de las soluciones.

### 5.1 Archivos fuente de soluciones propuestas

Todos los programas implementados en *Java* o en *Python*

Para el problema:

- Entregar un archivo de código fuente en *Java* (`.java`) o *python* (`.py`) con su código fuente de la solución que se presenta.
- Incluir como encabezado de cada archivo fuente un comentario que identifique el (los) autor(es) de la solución.
- Denominar `ProblemaP1.java` o `ProblemaP1.py` el archivo de la solución que se presente.

Nótese que, si bien puede utilizarse un *IDE* como *Eclipse* o *Spyder* durante el desarrollo del proyecto, la entrega requiere incluir solo un archivo por cada solución. El archivo debe poderse compilar y ejecutar independientemente (sin depender de ninguna estructura de directorios, librerías no estándar, etc.).

### 5.2 Archivos que documentan la solución propuesta

La solución al problema debe acompañarse de un archivo de máximo 3 páginas que la documente, con extensión `.pdf`. El nombre del archivo debe ser el mismo del código correspondiente (`ProblemaP1.pdf`).

Un archivo de documentación debe contener los siguientes elementos:

- 0 *Identificación*  
Nombre de autor(es)
- 1 *Algoritmo de solución*  
Explicación del algoritmo elegido. Si hubo alternativas de implantación diferentes, explicar por qué se escogió la que se implementó. Generar al menos una gráfica que apoye la explicación del algoritmo implementado. No se debe copiar y pegar código fuente como parte de la explicación del algoritmo.
- 2 *Análisis de complejidades espacial y temporal*  
Cálculo de complejidades y explicación de estas.

La nota del informe corresponde a un 50% de la nota total de la entrega del proyecto, solamente si se entrega el código fuente de la solución implementada. En caso de no entregar el código fuente, no se hará evaluación del informe y la nota del proyecto será cero.

Además de la pertinencia del texto como explicación de la solución implementada, se evaluará la calidad en la redacción del texto y en el diseño de las gráficas. Se evaluará también que la explicación del algoritmo integre los conceptos relacionados con las técnicas de diseño de algoritmos cubiertas en el curso.

Téngase en cuenta que los análisis de 2 tienen sentido en la medida que la explicación de 1 sea clara y correcta. No se está exigiendo formalismo a ultranza, pero sí que, como aplicación de lo estudiado en el curso, se pueda describir un algoritmo de manera correcta y comprensible.

### **5.3 Consideraciones sobre la entrega**

Lea bien las recomendaciones de entrada/salida. Su proyecto será evaluado, en gran parte, por una máquina: si en la ejecución del programa el formato de salida no coincide perfectamente con lo requerido, la calificación de la ejecución será cero, sin importar la cantidad de código que haya desarrollado.

La forma en que se presenta la documentación debe respetarse, aunque su evaluación no sea tan automática como la del software. Hay que nombrar los archivos como se espera que se nombren, comprimirlos como se pide que se compriman, etc. Cualquier desviación en cuanto a lo que se pide produce deméritos en la calificación final.

El software se evalúa desde la línea de comandos de Linux. No hay problema en desarrollar en cualquier otro sistema (Mac, Windows, ...) siempre que se produzca código estándar en java o python. Cada problema se debe resolver en un (1) archivo.