

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 eficiencia en tiempo y espacio.

1 CONDICIONES GENERALES

El proyecto se divide en tres partes independientes entre sí. Este documento describe la PARTE 2. 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

A Reconstruyendo un alfabeto de Pandora

Unos estudiantes de DALGO son invitados a un viaje intergaláctico al planeta Pandora. En el planeta encuentran un libro cuyas páginas se encuentran ordenadas y numeradas. ¡¡Después de semanas de análisis lingüistas expertos llegan a la conclusión que se trata de un diccionario!!, y más impresionante aún es que usa un alfabeto que es un subconjunto del alfabeto del castellano.

Problema

A pesar de que el alfabeto de Pandora es un subconjunto de nuestro alfabeto, el orden parece no ser el mismo. Acuden entonces a los estudiantes de DALGO para que identifiquen el orden correcto de las letras del alfabeto a partir de los contenidos de las páginas y su numeración.

Ejemplo:

Suponga que el diccionario tiene tres páginas con las siguientes cadenas de caracteres (puede interpretar una cadena de cada línea como una palabra):

Página 0	Página 1	Página 2
h hjh hjmh	hm j jjm	m mh mmhj

La salida para este ejemplo corresponde a “hjm” que es el único posible orden que valida las tres páginas como diccionario.

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 contiene varias líneas. La primera línea de cada caso contiene 2 enteros $n(1 \leq n), m(m \leq 10^3)$, n es el número de páginas, y m es el número de palabras presentes en cada página. Posteriormente va a encontrar n líneas donde se estipulan las palabras que contiene cada página. Cada una de estas líneas está compuesta de un entero $p_i (0 \leq p_i < n)$ que representa el número de la página y m cadenas de caracteres separadas por un espacio sencillo. Cada una de las m cadenas de caracteres de cada página se debe interpretar como una palabra. La longitud máxima de una palabra es 100.

Descripción de la salida

Para cada caso de prueba, imprimir el alfabeto ordenado de Pandora o la palabra “ERROR” si no es posible encontrar un orden válido del alfabeto.

Ejemplo de entrada / salida

Para el ejemplo realizado anteriormente, la descripción de la entrada/salida es la siguiente.

Entrada	Salida
4	hjm
3 3	xpjw
2 m mh mmhj	ERROR
0 h hjh hjmh	abmhnk
1 hm j jjm	
1 5	
0 xx xp pj jjj jjw	
1 4	
0 ab ac cc cb	
1 5	
0 ab ah an mn mk	

Nótese que las páginas no necesariamente son entregadas en orden.

Nota: Se van a diseñar casos de prueba para valores de n , y m muchos 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.

4 COMPRENSIÓN DE PROBLEMAS ALGORITMICOS

A continuación, se presentan un conjunto de escenarios hipotéticos que cambian el problema original. Para cada escenario debe contestar¹: (i) que nuevos retos presupone este nuevo escenario -si aplica-?, y (ii) que cambios -si aplica- le tendría que realizar a su solución para que se adapte a este nuevo escenario?

ESCENARIO 1: Suponga solo se sabe el orden para un subconjunto de páginas.

ESCENARIO 2: Suponga que existe la posibilidad que en el alfabeto de pandora se permitan los dígrafos como letras. Las famosas “ll” y “ch” que antes pertenecían a nuestro abecedario son ejemplos de dígrafos.

Nota: Los escenarios son independientes entre sí.

5 ENTREGABLES

El proyecto puede desarrollarse por grupos de hasta tres 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 `proyectoDalgoP2.zip`. Este archivo es una carpeta de nombre `proyectoDalgoP2`, 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 `ProblemaP2.java` o `ProblemaP2.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 (`ProblemaP2.pdf`).

¹ NO tiene que implementar la solución a estos escenarios, el propósito es meramente analítico.

Un archivo de documentación debe contener los siguientes elementos:

0 Identificación

Nombre de autor(es)

Identificación 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.

Deseable:

Anotación (contexto, pre-, poscondición, ...) para cada subrutina o método que se use.

2 Análisis de complejidades espacial y temporal

Cálculo de complejidades y explicación de estas. Debe realizarse un análisis para cada solución entregada.

3 Respuestas a los escenarios de comprensión de problemas algorítmicos.

Respuesta a las preguntas establecidas en cada escenario. NO tiene que implementar la solución a estos escenarios, el propósito es meramente analítico.

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.

No describa un algoritmo con código GCL a menos que lo considere necesario para explicarlo con claridad. Y, si lo hace, asegúrese de incluir aserciones explicativas, fáciles de leer y de comprender.