

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, eficiencia en tiempo y espacio.

1 CONDICIONES GENERALES

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

Una cantante muy famosa quiere aprovechar su éxito para crear un fondo que le permita desarrollar proyectos sociales. Utilizando una estrategia de mercadeo efectiva, logra conectarse con un grupo grande de personas que están dispuestas a aportar a este fondo. Sin embargo, debido a diferentes escándalos de corrupción que han ocurrido en el país, todas las personas interesadas le manifestaron que solamente aportarían su dinero si conocían a todas las demás personas que aportarían al fondo.

Problema

Dada la cantidad de dinero que aportaría cada una de las personas interesadas y la información de qué personas se conocen entre ellas, determinar cuánto dinero se podría recaudar como máximo para el fondo.

Ejemplo 1:

Dada la información de dinero a aportar y las relaciones entre las personas definida por la siguiente tabla:

Persona	Dinero (Dolares)	Conocidos
Ana	1000	Juan, Paola
Carlos	1500	Juan
Juan	700	Ana, Carlos, Sandra
Paola	1200	Ana, Sandra
Sandra	1400	Juan, Paola

La máxima cantidad de dinero que se puede recaudar es 2600 sumando los aportes de Paola y Sandra

Ejemplo 2:

Dada la información de dinero a aportar y las relaciones entre las personas definida por la siguiente tabla:

Persona	Dinero (Dolares)	Conocidos
Bernardo	800	Daniela, Ernesto, Miguel, Viviana
Daniela	1300	Bernardo, Miguel
Ernesto	900	Bernardo, Miguel, Viviana
Miguel	1100	Bernardo, Daniela, Ernesto
Viviana	1200	Bernardo, Ernesto

La máxima cantidad de dinero que se puede recaudar es 3200 sumando los aportes de Bernardo, Daniela y Miguel

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 inicia con una línea que tiene N números, los cuales representan la cantidad de dinero que aportaría cada una de las personas interesadas. Luego de esto hay N líneas que indica para cada persona los índices de las personas que conoce (entre 1 y N).

Tamaños de entrada: $1 \leq N \leq 10000$

Descripción de la salida

Para cada caso de prueba, imprimir la cantidad de dinero que se puede recaudar como máximo, seguido de los índices de las personas con quienes se lograría este recaudo. Los números reportados para cada caso de prueba deben estar separados por espacio

Ejemplo de entrada / salida

Para los ejemplos descritos arriba:

Entrada	Salida
2	2600 4 5
1000 1500 700 1200 1400	3200 1 2 4
3 4	
3	
1 2 5	
1 5	
3 4	
800 1300 900 1100 1200	
2 3 4 5	
1 4	
1 4 5	
1 2 3	
1 3	

Ejemplo con tres casos de prueba:

Entrada	Salida
3	2600 4 5
1000 1500 700 1200 1400	3200 1 2 4
3 4	2100 1 2 3
3	
1 2 5	
1 5	
3 4	
800 1300 900 1100 1200	
2 3 4 5	
1 4	
1 4 5	
1 2 3	
1 3	
600 700 800	
2 3	
1 3	
1 2	

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.

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: Las personas podrían aceptar no conocer máximo una de las personas que van a aportar.

ESCENARIO 2: Las personas solamente necesitarían conocer al menos una de las personas que van a aportar.

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 proyectoDalgoP3.zip. Este archivo es una carpeta de nombre proyectoDalgoP3, comprimida en formato .zip, dentro de la cual hay archivos fuente de soluciones propuestas y archivos que documentan cada una de las soluciones.

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

5.1 Archivos fuente de soluciones propuestas

Todos los programas deben ser 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 `ProblemaP3.java` o `ProblemaP3.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 (`ProblemaP3.pdf`).

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. 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. Argumentar si el algoritmo planteado resuelve perfectamente el problema.
- 2 *Análisis de complejidades espacial y temporal*
Cálculo de complejidades y explicación de estas.
- 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.

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.