

Trabajo Práctico 1

Fecha de entrega: domingo 3 de septiembre, hasta las 23:59 hs.

Fecha de reentrega: domingo 24 de septiembre, hasta las 23:59 hs.

Este trabajo práctico presenta un problema y deberá resolverse individualmente. Se pide encontrar una solución algorítmica al problema propuesto y desarrollar los siguientes puntos:

1. Describir detalladamente el problema a resolver dando ejemplos del mismo y sus soluciones.
2. Explicar de forma clara, sencilla, estructurada y concisa, las ideas desarrolladas para la resolución del problema. Para esto se pide utilizar pseudocódigo y lenguaje coloquial combinando adecuadamente ambas herramientas (**¡sin usar código fuente!**). Se debe también justificar por qué el procedimiento desarrollado resuelve efectivamente el problema.
3. Deducir una cota de complejidad temporal del algoritmo propuesto (en función de los parámetros que se consideren correctos) y justificar por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada.
4. Dar un código fuente claro que implemente la solución propuesta. El mismo no sólo debe ser correcto sino que además debe seguir las *buenas prácticas de la programación* (comentarios pertinentes, nombres de variables apropiados, estilo de indentación coherente, modularización adecuada, etc.).
5. Realizar una experimentación computacional para medir la performance del programa implementado. Para ello se debe preparar un conjunto de casos de test que permitan observar los tiempos de ejecución en función de los parámetros de entrada.

Respecto de las implementaciones, se acepta cualquier lenguaje que permita el cálculo de complejidades según la forma vista en la materia. Además, debe poder compilarse y ejecutarse correctamente en las máquinas de los laboratorios del Departamento de Computación. **Se debe incluir un script o Makefile que compile un ejecutable que acepte como entrada lo solicitado en el problema** La cátedra recomienda el uso de C++ o Java, y se sugiere consultar con los docentes la elección de otros lenguajes para la implementación. **Solo se permite utilizar 1 lenguaje para resolver el problema.** Se pueden utilizar otros lenguajes para presentar resultados.

La entrada y salida de los programas **deberá hacerse por medio de la entrada y salida estándar del sistema**. No se deben considerar los tiempos de lectura/escritura al medir los tiempos de ejecución de los programas implementados.

Deberá entregarse un informe impreso **con a lo sumo 10 paginas** que desarrolle los puntos mencionados. Por otro lado, deberá entregarse el mismo informe en formato digital acompañado de los códigos fuentes desarrollados e instrucciones de compilación, de ser necesarias. Estos archivos deberán enviarse a la dirección `algo3.dc@gmail.com` con el asunto *“TP 1: Apellido”*.

En la continuación de la película Tiempo de Valientes el Licenciado Mariano Silverstein se encuentra al frente de la brigada y debe detectar un grupo de agentes que siguen siendo fieles a Lebonian. Por lo tanto Mariano examina a los agentes con un procedimiento simple y efectivo que determina qué tan confiable, a la brigada, es un grupo de agentes.

Mariano determina la confiabilidad de un grupo de agentes utilizando una encuesta que mide la confianza entre los mismos, los agentes deben opinar sobre cada uno, inclusive ellos mismos. Como resultado de la encuesta, Mariano obtiene un conjunto de afirmaciones de la forma “X dice que Y es confiable”, “X dice que Y no es confiable”. Si X es confiable entonces Mariano asume que lo que el agente opina siempre es verdadero. Caso contrario, si X no es confiable, entonces las opiniones pueden ser verdaderas o falsas. Mariano quiere determinar la máxima cantidad de agentes en los que se puede confiar, de modo que las respuestas a la encuesta sean consistentes.

Como ejemplo, asumamos que hay cuatro informantes A,B,C y D, que respondieron “A dice que B es confiable pero D no lo es”, “B dice C no es confiable”, “C dice que A y D son confiables”. En este caso hay a lo sumo dos informantes de confianza.

Se debe escribir un programa que ayude a Mariano a determinar los agentes en que puede confiar. Es decir, que busque el conjunto más grande de agentes tal que:

- Ningún agente en el conjunto diga que es confiable un agente que no esta en el conjunto.
- Ningún agente en el conjunto diga que no es confiable un agente en el conjunto.

La entrada consiste de varios casos que se deben procesar en lote. Cada caso comienza con dos enteros no negativos $1 \leq i$ y $0 \leq a$, separados por espacios en blancos. i es el número de informantes y a es el número de respuestas a encuestas. Luego hay a líneas, cada una con dos números x y y ($1 \leq x \leq i$, $1 \leq |y| \leq i$) separados por espacios en blanco. Si y es positivo, la línea de entrada significa que el agente “ x dice que el agente y es confiable”. Si y es negativo entonces, la línea de entrada significa que el agente “ x dice que el agente $-y$ no es confiable”. El final de la entrada se indica con una línea con dos ceros. Por cada caso de entrada escribir en la salida una línea con la máxima cantidad de agentes.

Se debe analizar la complejidad temporal y expresarla en función de i y de a . Encontrar un algoritmo utilizando *backtracking* que tenga peor caso mejor que $\mathcal{O}(2^i i^3 a^3)$. Utilizar por lo menos dos podas para mejorar el algoritmo y evaluar experimentalmente como resultan. Encontrar en que tipo de casos las podas ayudan y en que casos donde no.

Entrada de ejemplo	Salida esperada para la entrada de ejemplo
4 5	2
1 2	0
1 -4	2
2 -3	
3 1	
3 4	
1 1	
1 -1	
3 3	
1 2	
2 3	
3 -1	
0 0	