

Tarea 1

Profesor: Diego Arroyuelo
 Ayudantes: Alan Grez, Sebastián Sepúlveda
 alan.grez@sansano.usm.cl,
 sebastian.sepulvedab@sansano.usm.cl

Fecha de Inicio: 25 de octubre, 2021
 Fecha de Entrega: 8 de noviembre, 2021
 Plazo máximo de entrega: 5 días.

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Pueden usarse los lenguajes de programación C, C++, Python, y Java.

1. Joins en Bases de Datos [60 %]

En bases de datos relacionales, la operación join (denotada con \bowtie) permite combinar tuplas de distintas relaciones, formando tuplas más grandes. Dado que la normalización en bases de datos relacionales genera varias relaciones, la operación join permite (entre otras cosas) obtener la información original (es decir, antes de la normalización). Esto hace a los joins una de las operaciones más típicas y empleadas en bases de datos relacionales. Por ejemplo, si tenemos las relaciones $R(A, B)$ (es decir, que tiene atributos A y B) y $S(B, C)$ (con atributos B y C), la operación $R(A, B) \bowtie S(B, C)$ produce como resultado una relación $T(A, B, C)$. En este caso, el atributo B es conocido como *atributo de join*: es ese atributo el que se usa para combinar tuplas de R y S . Vamos a considerar únicamente el caso de join natural en esta tarea. Esto es, un join que se realiza sólo teniendo en cuenta la igualdad de los valores de los atributos de join. Por cada tupla $(a_i, b_j) \in R$ y $(b_j, c_k) \in S$, el join produce la tupla (a_i, b_j, c_k) en T . Considere el siguiente ejemplo de join:

R	$(A \quad B)$		S	$(B \quad C)$		T	$(A \quad B \quad C)$
	1 2						
	2 4						
	4 1			3 1			1 2 3
	5 2	\bowtie		2 3	=		5 2 3
	4 5			5 3			4 5 3
	3 4						3 5 3
	3 5						

La operación join está definida no sólo para relaciones binarias (como lo hemos explicado), sino que puede extenderse a relaciones de cualquier aridad (cantidad total de atributos), y cualquier cantidad de atributos de join. Por ejemplo, podemos tener $R(A, B, C) \bowtie S(A, C, D)$, que producirá la relación $T(A, B, C, D)$. La tarea consiste en implementar un algoritmo de fuerza bruta para resolver la operación join. Para simplificar la operación, los atributos de cada relación serán representados por números enteros (en lugar de letras como hemos usado en los ejemplos).

Formato de Entrada

La entrada de datos será a través de la entrada estándar (**stdin**). La primera línea de la entrada contiene un único número entero N , indicando la cantidad de atributos de la relación R . Puede asumir que $1 \leq N \leq 100$. La siguiente línea de la entrada contiene N valores enteros, separados entre sí por un único espacio en blanco, indicando los atributos de la relación R . Los valores están ordenados de forma creciente. La siguiente línea contiene un valor entero NR , tal que $1 \leq NR \leq 10,000$, e indica la cantidad de tuplas de la relación R . Luego, le siguen NR líneas, cada una con N valores enteros (separados entre sí por un único espacio), correspondientes a las tuplas de la relación R . A continuación, le sigue una línea que contiene un único número entero M , indicando la cantidad de atributos de la relación S . Puede asumir que $1 \leq M \leq 100$. La siguiente línea de la entrada contiene M valores enteros, separados entre sí por un único espacio en blanco, indicando los atributos de la relación S . Los valores están ordenados de forma creciente. La siguiente línea contiene un valor entero NS , tal que $1 \leq NS \leq 10,000$, e indica la cantidad de tuplas de la relación S . Luego, le siguen NS líneas, cada una con M valores enteros (separados entre sí por un único espacio), correspondientes a las tuplas de la relación S .

Un ejemplo (que replica el ejemplo mostrado anteriormente) es el siguiente:

```
2
0 1
7
1 2
2 4
4 1
5 2
4 5
3 4
3 5
2
1 2
3
3 1
2 3
5 3
```

Esto significa que la relación R tiene atributos identificados con 0 y 1 (línea 2), mientras que la relación S tiene atributos 1 y 2 (línea 12). Para la operación $R \bowtie S$, el atributo de join es el 1.

Formato de Salida

La salida se hará a través de la salida estándar (**stdout**). Debe mostrar el resultado de la operación join sobre las dos relaciones dadas en la entrada. El formato es el siguiente: la primera línea debe contener un valor P , indicando la cantidad de atributos de la relación resultante del join. Luego, le sigue una línea que contiene P valores enteros, separados entre sí por un único espacio, y ordenados de forma creciente. Esos valores indican los atributos de la relación resultante. Luego, le sigue una línea que contiene un único valor entero NT , que es la cantidad de tuplas del join. Finalmente, se tienen NT líneas, cada una con P valores enteros separados por un único espacio, correspondientes a las tuplas del join.

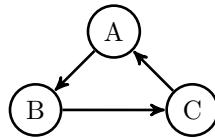
La salida correspondiente al ejemplo mostrado anteriormente es:

```
3
0 1 2
4
1 2 3
5 2 3
4 5 3
3 5 3
```

Eso significa que la relación resultante del join tiene atributos 0, 1, y 2.

2. Contando Triángulos en Grafos [40 %]

Sea $G = (V, E)$ un grafo dirigido con conjunto de vértices V y aristas E . En esta parte, debe implementar un algoritmo que use el algoritmo de join de la sección anterior para contar la cantidad de triángulos que tiene G . Esto es una aplicación importante en muchas aplicaciones, como por ejemplo redes sociales y biología. Un triángulo es un patrón del siguiente tipo:



Suponga que los arcos del grafo estarán representado por una relación binaria $E(A, B)$, en donde por cada arco $a \rightarrow b$ en el grafo, tenemos la tupla $(a, b) \in E$. El objetivo es expresar ese triángulo como una consulta de join.

La entrada de datos se hará mediante la entrada estándar (**stdin**). Cada línea contiene dos valores enteros, que representan un arco (dirigido) del grafo. la entrada es terminada con **EOF**. La salida se hará a través de la salida estándar (**stdout**). Se debe imprimir una única línea que contiene la cantidad de triángulos del grafo dado en la entrada.

3. Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea1-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda), o alternatively usando formato zip, en el sitio Aula USM del curso, a más tardar el día 8 de noviembre, 2021, a las 23:59:00 hrs (Chile Continental), el cual contenga:

- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, Nombre y ROL de cada integrante del grupo.
- `README.txt`, Instrucciones de compilación en caso de ser necesarias.
- `Makefile`, Instrucciones para compilación automática.