## **Evaluación asincrónica 4** de Estrategias Algorítmicas, ITESO, Luis Gatica

### Fecha de entrega: viernes 1 de diciembre de 2023

Nombre: Yael Alejandro Rodríguez Barreto

Esta tarea consiste en resolver problemas del tipo concurso de programación: uno general (para todxs) y algunos específicos (asignaciones al final del documento). El primer problema (sobre grafos) será tratado de una forma muy particular. Los últimos problemas (los específicos) serán tratados simplemente como problemas de concurso de programación, y serán asignaciones específicas a cada estudiante o pareja (asignaciones al final).

Es posible hacer el primer problema en parejas, pero los problemas específicos son de resolución individual. Puede entregarse todo en un solo documento y subirse a canvas una sola vez (por unx solx miembrx).

**Introducción al problema sobre grafos**

Usted está en un concurso de programación en Java. El problema que está resolviendo consiste en calcular la ruta más barata desde un nodo a todos los nodos de un grafo, pero cada vez que envía su solución obtiene un *TIME LIMIT EXCEEDED* o un MEMORY LIMIT EXCEEDED por parte del juez automático. Uno de sus amigos, infiltrado en la organización del concurso, le filtra el caso de prueba donde su programa tiene conflictos. Además, resulta que en realidad ese caso de prueba es el único, por lo que usted puede (y debe) optimizar su código específicamente para las restricciones de ese caso. ¡Gane el concurso!



**Descripción**

La entrada y la salida del caso de prueba corresponden a los archivos *t8.in* y *t8.out* respectivamente.

**Entrada**

La primera línea contiene dos enteros: *N* y *M*. *N* es el número de vértices del grafo y *M* es el número de aristas.

Para las siguientes *N* líneas, la línea *i* contiene el nombre del i-ésimo vértice en una cadena de cuatro caracteres (todos mayúsculas del alfabeto inglés).

Para las siguientes *M* líneas, la línea *j* contiene una arista: el nombre del vértice de origen, un espacio, el nombre del vértice destino, un espacio, el peso del arista (*w* <= 1000).

La siguiente línea contiene el nombre del vértice de inicio (*s*) para calcular las rutas más cortas.

**Salida**

La salida consiste en *N* líneas. La línea *i* contiene el nombre del i-ésimo vértice (siguiendo el mismo orden de la entrada) seguido de dos puntos (':'), un espacio y un entero *di*: la distancia mínima de *s* al vértice *i*. Por supuesto, cuando *i* == *s*, se cumple que *di* == 0.

Para comprobar que su programa es suficientemente eficiente, pruébelo en la plataforma de *omegaup*. (la liga será cargada a canvas; el concurso aparecerá como “ya terminado”: use el modo de práctica para enviar soluciones). La clase principal será llamada *Main* (lea los **lineamientos** en el documento de la *Compilación de problemas de concurso*). El tiempo y la memoria permitidos para una solución están limitados. Complete su tarea escribiendo un párrafo para justificar su solución (el algoritmo o la *estrategia algorítmica* seguida) y escribiendo sus **conclusiones generales del semestre** (aprendizajes logrados, retos satisfechos, dudas que queden sueltas…).

[Nota: usted/es no está/n compitiendo contra sus compañerxs. Su nota depende de su propia solución y desempeño, que no serán comparados con los del resto.]



|  |  |
| --- | --- |
| Criterio de evaluación | Puntos |
| Programación correcta de la solución al problema de grafos | 35 |
| Tiempo de ejecución | 5 |
| Memoria | 5 |
| Captura de pantalla de la aceptación por parte del juez de *omegaup* | 5 |
| Justificación de su solución | 5 |
| Conclusiones generales del semestre | 15 |
| Problema específico | 30 |
| NOTA MÁXIMA POR CADA ENTREGABLE | **100** |

**Consideraciones:**

* El documento con el *banco de problemas* está aparte.
* Para tener los puntos de los problemas específicos es imprescindible incluir **captura de pantalla de salida del juez** donde sea visible cuántos casos de prueba fueron resueltos o, en su defecto, el tipo de error que el código arrojó (tiempo, memoria, formato, respuesta incorrecta...).
* Problemas “extra” no serán considerados para calificación.
* Lxs estudiantes asignadxs a los mismos problemas no necesariamente son equipo. Simplemente cada unx debe entregar su propia solución a esos problemas. (Esto aplica sólo si no hay suficientes problemas para asignar problemas distintos a todxs.)
* El plagio deja a las entregas implicadas con calificación 0.

**RODRIGUEZ BARRETO YAEL ALEJANDRO**

* Subsecuencias distintas

**Subsecuencias distintas**

Una subsecuencia de una secuencia dada S consiste en S con cero o más elementos eliminados. Formalmente, una secuencia *Z* = *z*1*z*2…*zk* es una subsecuencia de *X* = *x*1*x*2…*xm* si existe una secuencia estrictamente creciente *i*1, *i*2..., *ik* de índices de *X* tal que para todo *j* = 1, 2, . . . , *k*, tenemos *xi*[*j*] = *zj*. Por ejemplo, *Z* = *bcdb* es una subsecuencia de *X* = *abcbdab* con la correspondiente secuencia de índices: 2, 3, 5, 7. Tu trabajo es escribir un programa que cuente el número de ocurrencias de *Z* en *X* como una subsecuencia, tal que cada una tiene una secuencia de índices distinta.

**Input**

La primera línea contiene un entero *N* indicando el número de casos de prueba. La primera línea de cada caso de prueba contiene una cadena *X*, compuesta enteramente de caracteres alfabéticos en minúscula y con una longitud no mayor que 10,000. La segunda línea contiene otra cadena Z de longitud no mayor que 100 y también compuesta sólo de caracteres alfabéticos en minúscula.

**Output**

Por cada caso de prueba, muestra el número de ocurrencias distintas de *Z* en *X* como una subsecuencia. La salida de cada conjunto de entrada debe estar en una línea separada (terminada con salto de línea).

**Sample Input**

2

babgbag

bag

rabbbit

rabbit

**Sample Output**

5

3

**ASIGNACIONES:**

**ACOSTA MENDOZA LUIS RAUL**

* Estantería de Babel

**AMADOR GUDIÑO EDGAR EDUARDO**

* Noticias falsas

**ARELLANO VALDEZ ANA CAROLINA**

* La florería

**CHAVEZ CASTAÑEDA JOSE FERNANDO**

* Justicia laboral

**CORTEZ NAVARRO JUAN PABLO**

* Productos más vendidos

**COTA SAMAUE JOSE ANDRES**

* Formación por estaturas

**DE LA CRUZ LOPEZ ISAAC**

* Koopedo

**DIAZ CAMPOS JOSE JUAN**

* Minúsculas

**DIAZ DE LEON GUIZAR CRISTIAN MIGUEL**

* Que no se te vaya el tren

**GARCIA GARCIA PERLA MARINA**

* Diámetro de redes sociales

**OROZCO SANCHEZ DIEGO ARTURO**

* Gastos perplejos

**ORTEGA ANGUIANO FRANCISCO AARON**

* Secuencias

**PATRICIO MARTINEZ EMMANUEL**

* Permutación común

**PENICHE BARBA TIRZAH**

* Sucursales convenientes

**QUINTERO VALADEZ SAMANTHA ABIGAIL**

* Bolitas de amigas

**RENTERIA CAMACHO ALBERTO**

* ADN

**RIZO VELARDE DION**

* Utilidades

**RODRIGUEZ BARRETO YAEL ALEJANDRO**

* Subsecuencias distintas

**RUELAS GAYTAN LORENA**

* Ruta crítica

**TORRES DUEÑAS MIGUEL ALBERTO**

* Multinivel

**VALADEZ FERNANDEZ PAULINA**

* Estaciones de ser vicio

**VERGARA ROMO ARTURO BENJAMIN**

* Los caprichos del chef

**ZARATE SOLANO BENJAMIN LEONARDO**

* Lista telefónica