

Examen 1 Septiembre – Diciembre 2018

Todos los caminos conducen a Roma

1 Introducción

Existe un antiguo dicho que “Todos los caminos conducen a Roma.” Si fuera verdad, existe un sencillo algoritmo para encontrar una ruta entre cualquier par de ciudades. Para ir de una ciudad A a una ciudad B, un viajero podría tomar un camino de A a Roma, y luego de Roma a B. Por supuesto, una ruta más corta puede existir.

La red de caminos en el Imperio Romano tenía una estructura sencilla: comenzando en Roma, un número de caminos se extendían a las ciudades cercanas. Desde estas ciudades, más caminos se extendían a las siguientes ciudades más lejanas, y así sucesivamente. Así, las ciudades podían pensarse como existiendo en niveles alrededor de Roma, con las ciudades en el i -ésimo nivel solo conectadas con ciudades en el $(i - 1)$ -ésimo nivel y el $(i + 1)$ -ésimo nivel; Roma se consideraba ser el nivel 0.

No existían ciclos en la red de caminos. Cualquier ciudad en el nivel i estaba conectada con una sola ciudad en el nivel $i - 1$, pero estaba conectada a cero o más ciudades en el nivel $i + 1$. Así, para llegar a Roma de una ciudad dada en el nivel i , un viajero podía simplemente caminar por el único camino que llevaba a la ciudad conectada en el nivel $i - 1$, y repetir este proceso, cada paso sucesivo llevándolo más cerca de Roma.

2 Requerimientos del programa

Dado una red de caminos y ciudades, su tarea es encontrar la ruta más corta entre cualesquiera dos ciudades, donde la distancia se mide en el número de ciudades intermedias.

El programa se debe poder ejecutar desde la consola con el siguiente comando:

```
java CaminoRomano <archivo>
```

donde <archivo> es el archivo de casos de prueba. El archivo podrá contener varios casos de prueba.

2.1 Formato del Archivo de entrada

- La primera línea es el número de casos de prueba
- Seguidos por una línea en blanco
- La primera línea de cada caso del archivo contiene dos números, m y n , en notación decimal separados por un solo espacio
 - El primer número (m) es el número de caminos en la red de caminos a considerar
 - El segundo número (n) representa el número de consultas que siguen más adelante en el archivo.
- Para cada caso de prueba
 - Las siguientes m líneas en la entrada contienen los nombres de un par de ciudades separadas por un solo espacio. Los pares de nombres indican que un camino conecta las dos ciudades

nombradas. La primera ciudad en una línea existe en un nivel más bajo que la segunda ciudad nombrada. Para cada caso de prueba, no existen dos líneas de entrada que se repitan.

- Las siguientes n líneas, para cada caso de prueba en la entrada, cada una contiene los nombres de pares de ciudades separadas por un solo espacio. Estos son los pares a consultar. Su tarea es, para cada par de consulta, encontrar el camino más corto desde la primera ciudad nombrada a la segunda. Cada una de las ciudades en el par está garantizada a haber aparecido en algún lugar de la sección anterior, para cada caso de prueba, describiendo la estructura de los caminos.
- Cada caso de prueba estará separado con una línea

El nombre de una ciudad se compone de una o más letras, la primera de las cuales estará en mayúscula. No existen dos ciudades que empiecen con la misma letra. El nombre de Roma siempre aparece al menos una vez en esta sección de la entrada, para cada caso de prueba; esta ciudad se considera el nivel 0, el nivel de número más bajo.

2.1.1 Ejemplo de Archivo

1

7 3

Roma Torino

Torino Venecia

Torino Genova

Roma Pisa

Pisa Florencia

Venecia Atenas

Torino Milan

Torino Pisa

Milan Florencia

Atenas Genova

2.2 Formato de Salida

En cada caso de prueba, para cada uno de los n pares a consultar, imprima una secuencia de letras mayúsculas indicando la ruta más corta entre las dos ciudades del par a consultar. La secuencia debe ser devuelta como letras consecutivas, sin espacio intermedio, en una sola línea.

Para cada caso de prueba, la primera línea de salida corresponde al primer par a consultar, la segunda línea de salida corresponde al segundo par a consultar, y así sucesivamente. Las letras en cada secuencia indican la primera letra de las ciudades en la ruta deseada entre las ciudades del par a consultar, incluyendo las ciudades consultadas en sí. Una ciudad nunca se podrá emparejar contra sí misma en una consulta. Imprima una línea en blanco entre las salidas para dos casos de prueba consecutivos.

2.2.1 Ejemplo de Salida

TRP

MTRPF

AVTG

3 Requerimientos de la entrega

La entrega se realizará a través del Moodle antes de finalizar la hora de laboratorio

4 Evaluación

El examen tiene una ponderación de 10 puntos. Se asignarán

- 4 puntos por código
 - 1 punto por modificar un algoritmo de búsqueda en grafos para utilizar la filosofía Romana
 - 1 punto por su llamada al algoritmo de búsqueda
 - 1 punto por su método de lectura del archivo
 - 1 punto por su programa principal
- 6 puntos por ejecución
 - 5 puntos por encontrar correctamente el camino más corto en todos los casos
 - 1,5 puntos por poder hacerlo con los casos de ejemplo dados
 - 3,5 puntos por poder hacerlo con otros casos
 - 1 punto por imprimir en el formato correcto

El programa debe correr sin errores.