

**Árboles y Grafos, 2024-2**  
**Proyecto Final**  
**Facultad de Ingeniería y Ciencias**  
**Pontificia Universidad Javeriana Cali**

---

### Instrucciones Generales

El proyecto consiste en diseñar y analizar una solución al problema que se plantea en las siguientes páginas y enviar dicha solución a través de la arena de programación. Además, debe elaborar un documento tipo informe donde explique su solución. A continuación se indican las reglas para la presentación del proyecto:

- El proyecto se debe realizar de forma completamente **individual** y no se puede usar en su desarrollo ningún tipo de ayuda de otras personas, de internet o de inteligencia artificial. No se debe buscar el problema en ninguna plataforma ni buscar soluciones o casos de prueba. No es posible tener conversaciones o reuniones con compañeros del curso ni con otras personas. Cualquiera de estas acciones se entenderá como una falta a las reglas del proyecto y al código de honor del curso.
- En la solución al problema debe leer los datos de entrada de la forma en la que se indica en el enunciado y debe imprimir los resultados con el formato allí indicado. No debe agregar mensajes ni agregar o eliminar datos en el proceso de lectura. La omisión de esta indicación puede generar que su programa no sea aceptado en la arena de programación.
- Puede resolver el problema en C/C++ o Python.
- El proyecto tendrá dos entregas que serán descritas a continuación:
  - **Entrega 1 (20%):** En esta entrega se debe elaborar un informe en formato **pdf** en el que se plasmen los siguientes elementos:
    - \* Descripción general del problema en sus propias palabras.
    - \* Especificación del problema.
    - \* Definición de 3 casos de prueba inventados y el resultado esperado para cada caso.
    - \* Explicación general de algunas ideas sobre la solución al problema donde indique aspectos que crea que deben aún ser perfeccionados y aspectos que aún no tenga claro cómo podrían realizarse.

En esta entrega puede adjuntar cualquier avance que tenga de la implementación. La intención es que a partir de lo indicado en el documento pueda recibir algún tipo de retroalimentación del profesor que le pueda ayudar a avanzar en la solución del problema.

El plazo máximo para enviar esta entrega por Discord es el día **6 de noviembre a las 23:59**.

- **Entrega Final (80%):** En esta entrega deberá enviar su solución a través de la arena de programación. Además, deberá enviar el reporte final del proyecto en el que debería haber realizado las correcciones indicadas para la entrega 1 y además adicionar los siguientes elementos:
  - \* Explicación general de la estrategia de solución y del proceso con el que se obtiene la respuesta para un caso de prueba inventado que sea lo suficientemente representativo.
  - \* Análisis de la complejidad de su solución.

El plazo máximo para enviar esta entrega por Discord y por la arena de programación es el día **25 de noviembre a las 07:59**. Ese mismo día y los días **26 y 27 de noviembre** se realizará la sustentación.

- La nota final del proyecto dependerá de la sustentación. Después de la sustentación se le asignará un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de la sustentación. Su nota definitiva será la nota obtenida con las ponderaciones indicadas anteriormente, multiplicada por ese valor. Si

su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.

Tenga muy en cuenta esta aclaración. El propósito de la sustentación es que cada estudiante demuestre que elaboró completamente. Por esta razón, es necesario trabajar a conciencia y preparar muy bien su sustentación. Durante la sustentación se harán preguntas sobre los detalles del proyecto y es posible que sea necesario explicar su implementación en el tablero.

- Este documento será actualizado próximamente para detallar cómo serán distribuidos los porcentajes en la entrega final.
- Las fechas planteadas para la entrega final y la sustentación son las fechas más adelante en las que es posible calificar y por ende **no será posible ningún tipo de extensión.**

## A - Zlatan Taxes

Source file name: `taxes.py`

Time limit:  $x$  seconds

In the Amicable, Great, Respected, and Advanced Empire of Zlatan, known as the AGRA Empire, there are multiple cities connected by cutting-edge roads where cars travel in both directions. Each road connects only two cities, and no two different roads connect the same pair of cities. Zlatan, with his somewhat old-fashioned view of the economy, has decided that each city must pay a fixed tax, which may vary depending on the size and population of the city. Additionally, since roads require constant monitoring for safety and ongoing maintenance, Zlatan decreed that the empire needs an additional fee for each road. However, being the generous and kind ruler that he is, Zlatan decided that for each road, only one of the two connected cities must pay the fee. This means that if two cities,  $u$  and  $v$ , are connected by a road, either city  $u$  or city  $v$  (but not both) will be responsible for paying the additional tax.

The tax for each road will be equal to  $l * k$ , where  $l$  is the length of the road, and  $k$  is the number of city pairs that will be disconnected if Zlatan decides to close the road.

Each city must then pay the value of the general tax assigned to it, as well as the tax for each road assigned to it for payment.

The mayors of each city decided to collaborate in order to meet the tax requirements in such a way that the maximum total tax paid by any city is as small as possible. To achieve this, they have reached out to Kenny, a skilled programmer from PUJC, who has recently learned about graph algorithms. Kenny's task is to help determine this value by means a computer program. In this story, you are Kenny.

### Input

The input may contain multiple test cases.

The first line of each test case contains two integers  $n$  ( $2 \leq n \leq 10000$ ) and  $m$  ( $0 \leq m \leq 20000$ ), where  $n$  denotes the number of cities and  $m$  denotes the number of roads respectively. The next line contains  $n$  space separated integers between 1 and 10000 (inclusive), where the  $j$ -th integer stands for the fixed tax of the  $j$ -th city. Each of the next  $m$  lines contains three integers  $u, v, l$  ( $1 \leq u, v \leq n, 1 \leq l \leq 10000, u \neq v$ ), denoting that there is a road between cities  $u$  and  $v$  whose length is  $l$ . There is an empty line between two consecutive test cases.

*The input must be read from standard input.*

### Output

For each case, print the case number and a message indicating the result according to the next format:

Case  $\#x$ : The highest tax that must be paid to Emperor Zlatan is  $y$ .

Print a blank line between two successive test cases.

*The output must be written to standard output.*

**Sample Input**

```
16 21
4 5 6 3 4 40 5 3 24 5 3 4 2 41 9 5
1 2 3
1 4 9
2 3 5
3 4 2
2 5 6
2 6 4
3 6 1
5 7 3
5 8 2
7 8 8
6 9 23
9 11 1
9 10 2
9 13 85
10 11 4
11 12 2
10 12 3
6 14 32
14 15 8
14 16 3
15 16 6

2 1
5 10
1 2 10

6 6
10 20 30 40 50 60
1 2 1
2 3 1
1 3 1
1 4 6
1 5 6
4 6 2

3 1
10 20 30
2 3 10
```

**Sample Output**

```
Case #1: The highest tax that must be paid to Emperor Zlatan is 1289.
Case #2: The highest tax that must be paid to Emperor Zlatan is 15.
Case #3: The highest tax that must be paid to Emperor Zlatan is 80.
Case #4: The highest tax that must be paid to Emperor Zlatan is 30.
```