

**Árboles y Grafos, 2025-1**  
**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 **12 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.
  - \* Descripción de los cambios que se hicieron en la solución desde la primera versión hasta la entrega final.
  - \* Análisis de la complejidad de su solución.
- 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 proximamente 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 Galactic Summit

Source file name: `summit.py`

Time limit: x seconds

In the Galactic Empire of Zlatan, the civilizations are distributed across a vast network of planetary colonies arranged in an  $N \times N$  grid. Each cell of the grid represents a city with an associated energy cost required for staying there and leaving.

After years of interstellar negotiations, Zlatan has decided to host a *Galactic Peace Summit*, bringing together the leaders of  $F$  allied civilizations. Each leader is currently located in a different city of the grid.

Interplanetary transport between two adjacent sectors (north, south, east, or west) consumes an amount of energy equal to the cost of every sector being traversed. For safety and logistical reasons, no leader can make more than  $T$  moves to reach the chosen meeting point.

Zlatan wants to select the hosting city so that the **total energy cost** of all leaders' journeys is as small as possible, and every leader can reach the meeting city without exceeding  $T$  moves. The cost of the energy in the hosting city will be on Zlatan.

Your task is to help the Imperial Council of Engineering determine the optimal sector for holding the Galactic Summit.

### Input

The input contains multiple test cases. Each test case begins with three integers  $N$ ,  $F$ , and  $T$ , where:

- $1 \leq N \leq 100$  is the size of the grid.
- $1 \leq F \leq 10$  is the number of leaders (friends of Zlatan).
- $1 \leq T \leq 100$  is the maximum number of allowed moves per leader.

The next  $N$  lines each contain  $N$  integers representing the energy cost of each sector. Then, a line follows with  $F$  pairs of integers  $(x_i, y_i)$  indicating the starting positions of the leaders.

The input terminates with a line containing three zeros.

*The input must be read from standard input.*

### Output

For each test case, print a single line describing the outcome of the search:

- If a suitable sector exists, display the coordinates of the chosen meeting sector  $(r, c)$  and the minimum total energy cost in the following format:

The Galactic Summit will be held at sector  $(r, c)$  with total energy cost = X

- If no such sector exists within the move limit  $T$ , print:

Zlatan is disappointed

If there is more than one suitable sector, the one that is lexicographically greater should be chosen.

*The output must be written to standard output.*

**Sample Input**

```
3 2 3
1 2 3
2 1 3
3 3 1
1 1 3 3
4 3 3
4 5 10 20
40 30 40 10
18 53 4 32
52 37 42 43
1 1 1 4 3 4
4 3 2
4 5 10 20
40 30 40 10
18 53 4 32
52 37 42 43
1 1 1 4 3 4
0 0 0
```

**Sample Output**

```
The Galactic Summit will be held at sector (3,2) with total energy cost = 5
The Galactic Summit will be held at sector (1,4) with total energy cost = 61
Zlatan is disappointed
```