

1. **(25 pts)** El algoritmo de Floyd-Warshall recibe un grafo y devuelve una matriz de distancias mínimas entre todos los pares de nodos. Proponga una modificación al algoritmo de Floyd-Warshall que permita determinar el camino más corto entre dos nodos dados sin cambiar su complejidad. **Explique** cómo modificar el algoritmo y proporcione un ejemplo. Puede mostrar el código modificado o usar palabras para explicar la modificación.
2. **(25 pts)** El profesor Adam tiene tres hijos que, lamentablemente, no se llevan bien. El problema es tan grave que los tres viven en casas separadas y asisten a escuelas distintas. Además, se niegan a caminar juntos hacia la escuela y cada uno se rehúsa a transitar por cualquier calle por la que el otro haya pasado ese día. Los niños no tienen problema en que sus caminos se crucen en una esquina. Afortunadamente, tanto las casas como las escuelas están en esquinas, pero más allá de eso, el profesor no está seguro de si será posible que los tres puedan asistir a la misma escuela. El profesor tiene un mapa de la ciudad. **Explique** cómo formular el problema de determinar si los hijos pueden llegar a las escuelas como un problema de flujo máximo. **La explicación debe ser clara y concisa; de lo contrario, se podría perder el punto.**
3. **(50 pts)** Proponga una implementación (Java, Python) para la función `flatopia` que resuelva un caso del problema de las **Autopistas**. La función recibe como entrada el número de ciudades, las coordenadas de cada ciudad, el número de autopistas y, finalmente, una lista de las autopistas ya construidas. La función debe resolver el caso, retornando una lista de autopistas que deben construirse para conectar todas las ciudades. Si no es necesario construir nuevas autopistas, retorne una lista vacía.

```
flatopia(number_of_cities: int, cities: list[(int, int)],  
         number_of_highways: int, highways: list[(int, int)]) -> list[(int, int)]
```

Para la implementación, puede suponer que ya ha implementado la estructura de Union-Find. Después de construir el grafo, la complejidad de la solución no debe superar  $O(E \log(E))$ ; **de lo contrario, no se validará el punto.**

## Autopistas

La nación isleña de Flatopia es completamente plana. Desafortunadamente, Flatopia tiene un sistema de autopistas públicas muy deficiente. El gobierno de Flatopia es consciente de este problema y ya ha construido una serie de autopistas que conectan algunas de las ciudades más importantes. Sin embargo, aún hay algunas ciudades a las que no se puede llegar a través de una autopista. Es necesario construir más autopistas para que sea posible conducir entre cualquier par de ciudades sin salir del sistema de autopistas.

Las ciudades de Flatopia están numeradas del 1 al  $N$  y la ciudad  $i$  tiene una posición dada por las coordenadas cartesianas  $(x_i, y_i)$ . Cada autopista conecta exactamente dos ciudades. Todas las autopistas (tanto las originales como las que se construirán) siguen líneas rectas, y por lo tanto, su longitud es igual a la distancia cartesiana entre las ciudades. Todas las autopistas se pueden usar en ambas direcciones. Las autopistas pueden cruzarse libremente entre sí, pero un conductor solo puede cambiar de autopista en una ciudad que esté en el extremo de ambas autopistas.

El gobierno de Flatopia quiere minimizar el costo de construir nuevas autopistas. Sin embargo, quieren garantizar que cada ciudad sea alcanzable por autopista desde cualquier otra ciudad. Dado que Flatopia es tan plana, el costo de una autopista es siempre proporcional a su longitud. Por lo tanto, el sistema de autopistas menos costoso será aquel que minimice la longitud total de las autopistas nuevas.

# Entrada

La primera línea de la entrada consiste en un número entero que indica el número de casos de prueba. Luego hay una línea en blanco, y los conjuntos de datos están separados por una línea en blanco.

Cada caso de prueba consta de dos partes. La primera parte describe todas las ciudades en el país, y la segunda parte describe todas las autopistas que ya se han construido.

La primera línea del caso de prueba contiene un único entero  $N$  ( $1 \leq N \leq 750$ ), que representa el número de ciudades. Las siguientes  $N$  líneas contienen cada una dos enteros,  $x_i$  y  $y_i$ , separados por un espacio. Estos valores dan las coordenadas de la ciudad  $i$ -ésima (para  $i$  de 1 a  $N$ ). Las coordenadas tendrán un valor absoluto no mayor a 10000. Cada ciudad tiene una ubicación única.

La siguiente línea contiene un único entero  $M$  ( $0 \leq M \leq 1000$ ), que representa el número de autopistas existentes. Las siguientes  $M$  líneas contienen cada una un par de enteros separados por un espacio. Estos dos enteros representan un par de números de ciudades que ya están conectadas por una autopista. Cada par de ciudades está conectado por a lo sumo una autopista.

# Salida

Para cada caso de prueba, escribe en el archivo de salida una línea por cada nueva autopista que debería construirse para conectar todas las ciudades con la menor longitud total posible de autopistas nuevas. Cada autopista debe ser presentada imprimiendo los números de las ciudades que conecta, separados por un espacio.

Si no es necesario construir nuevas autopistas (todas las ciudades ya están conectadas), el archivo de salida debe contener una línea con la frase ‘No new highways need’. Imprime una línea en blanco entre los casos de prueba.

## Ejemplo de Entrada

```
1
9
1 5
0 0
3 2
4 5
5 1
0 4
5 2
1 2
5 3
3
1 3
9 7
1 2
```

## Ejemplo de Salida

```
1 6
3 7
4 9
5 7
8 3
```