

Quiz I (25 %)- Laboratorio de Algoritmos y Estructuras III

Julio Perez

Ayudando a Ana

Contexto

Ana, una estudiante de Ingeniería de Producción en la Universidad Simón Bolívar, necesita llegar al comedor de la Casa del Estudiante después de su clase en el edificio de Aulas. Debido a las protestas en el campus, busca evitar las zonas bloqueadas. Ana confía en que su amigo, un estudiante en Ingeniería en Computación, pueda diseñar un algoritmo para encontrar la ruta más corta y segura.

Objetivo

Diseñar un algoritmo que, dado un mapa de la universidad y la ruta de las protestas, encuentre el camino más corto desde el punto de partida hasta el objetivo, evitando las zonas de protesta. Se asume lo siguiente:

- Las protestas siguen un patrón circular regresando al punto de inicio.
- Los estudiantes y las protestas tardan la misma cantidad de tiempo para ir de un sitio a otro.
- Todos los recorridos de un sitio A a un sitio B duran lo mismo.
- No se puede retornar al mismo sitio
- Los caminos pueden ser recorridos en ambos sentidos
- Las protestas no usan los caminos del mapa sino caminos verdes”

Ejemplo

Mapa de la universidad:

[[0, 1], [1, 2], [2, 4], [4, 10], [0, 3], [3, 4], [3, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10]]

Inicio: 0, Fin: 10, Ruta de protesta: [1, 2, 4, 3, 0]

En este ejemplo, el camino más corto sería: **(0, 3, 4, 10)**. La ruta que sigue la protesta es 1, 2, 4, 3, 0 y luego se devuelve al punto 1.

Implementación

Diseñar una clase **UniversityPathFinder** que contenga un método **find_shortest_path** para encontrar todos los caminos más cortos desde el punto de inicio hasta el destino, evitando las zonas de protesta. Si no hay camino posible, se debe devolver un set vacío.

Entrada: Un entero que indica el punto de inicio, un entero indicando el punto final, una lista de listas de enteros representando los caminos de una universidad y una lista representando la ruta de la protesta.

Salida: Un set de tuplas representando los caminos mas cortos o un set vacío si no es posible ir del inicio hasta el fin.

Evaluación

La clase será instanciada de la siguiente manera para la evaluación:

```
uf = UniversityPathFinder()
uf.find_shortest_path(start, end, university_map, protest_route)
```

Análisis

Se debe proporcionar la complejidad de la solución en el *Readme* y justificarla.

Formato de los Archivos de Prueba

Los archivos de prueba se encuentran en el archivo `test.zip`. Cada archivo de prueba sigue el siguiente formato:

- La primera línea contiene el número de caminos n de la universidad.
- Las siguientes n líneas contienen dos enteros cada una, que representan un camino entre dos sitios.
- La línea siguiente contiene la ruta de protesta, representada por una secuencia de enteros separados por espacios.
- La última línea contiene dos enteros que representan el nodo de inicio y el nodo de fin.
- Las líneas siguientes son las soluciones. Si no hay líneas, significa que no hay solución (conjunto vacío).

Ejemplo de Archivo de Prueba

```
12
0 1
1 2
2 4
0 3
3 4
3 5
5 6
6 7
7 8
8 9
9 10
4 10
1 2 4 3 0
0 10
0 1 2 4 10
```

Este archivo de prueba representa una universidad con 12 caminos, una ruta de protesta que pasa por los nodos 1, 2, 4, 3, y 0, y se busca el camino más corto desde el nodo 0 al nodo 10. Las soluciones aparecen después de la línea del nodo de fin. Por cada línea tenemos una secuencia de enteros representando un camino más corto separado por espacios. Si no hay soluciones, el archivo no tendrá líneas adicionales.

El tiempo máximo para completar el quiz es de 90 minutos.