

CI-2693: Laboratorio de Algoritmos y Estructuras 3

20 de junio de 2024

Objetivos

El objetivo de este laboratorio es que los estudiantes apliquen y comprendan en profundidad los algoritmos de Prim y Kruskal para la resolución de problemas de grafos.

Instrucciones Generales

- La solución debe ser subida a un repositorio privado en GitHub e incluir un archivo `README.md` con una breve descripción del proyecto.
- Deben escribir casos de prueba para cada algoritmo utilizando `unittest` o `pytest`. Referencia
- El código debe estar adecuadamente comentado utilizando docstrings y debe especificar el tipo de entrada y salida en los argumentos de cada función.
- Cada equipo debe estar conformado por 2 a 3 integrantes.

Ejercicio

Una nueva compañía de telecomunicaciones ha llegado a Caracas y desea ofrecer el mejor servicio de internet posible. Para lograrlo, debe ser extremadamente cuidadosa con sus gastos, específicamente al conectar todos los conjuntos residenciales con la central de servicios. La compañía recurre al Departamento de Computación de la USB, el cual les informa que ustedes están aprendiendo estos algoritmos y podrían ayudarles. Su tarea consiste

en, dada una lista de coordenadas que representan los conjuntos residenciales en un mapa de la ciudad, determinar la mínima cantidad de cable necesario para conectarlos y especificar qué conjuntos deben unirse. Pueden asumir que siempre es posible conectar dos conjuntos residenciales en línea recta mediante túneles subterráneos.

- **Entrada:** Una lista de tuplas que representan las coordenadas de los conjuntos residenciales (x, y) , donde x indica la distancia en km al este y y indica la distancia en km al norte, tomando como referencia que $(0, 0)$ es el punto más al oeste y al sur de la ciudad. Además, una tupla adicional representará las coordenadas de la central de servicios.
- **Salida:** La mínima cantidad de cable necesaria y el árbol de expansión mínima resultante M^* .
- **Requerimientos:**
 - Implementar los algoritmos de Prim y Kruskal para resolver el problema.
 - Verificar que las implementaciones sean correctas mediante casos de prueba.

Entrega

El código debe ser subido a GitHub antes del **27 de junio a las 10:30 am**. En el archivo `README.md`, deben explicar y argumentar la complejidad de su código.