

Programación III - Universidad Nacional de General Sarmiento

Trabajo Práctico 2: Temible operario del recontraespionaje

Tenemos un conjunto de espías en territorio enemigo, y necesitamos enviarles un mensaje a todos ellos. La comunicación se realiza de manera personal entre cada par de espías: se encuentran en un punto preacordado y se pasan el mensaje en un papel que se autodestruye luego de unos segundos. No todos los pares de espías pueden encontrarse, y representamos esta situación por medio de un grafo $G = (V, E)$ con un vértice por cada espía y una arista por cada par de espías que pueden encontrarse. Para cada arista $ij \in E$, tenemos además la probabilidad $p_{ij} \in [0, 1]$ de que el enemigo intercepte a los espías durante el encuentro y se arruine el operativo.

Para enviarles el mismo mensaje a toda nuestra red de espías, buscamos un árbol generador de G y los mensajes se transmiten a lo largo de las aristas de este árbol generador. Un *cuello de botella* de un árbol generador es su arista de mayor peso. En nuestro caso, buscamos un árbol generador con el cuello de botella más pequeño posible (es decir, buscamos minimizar el riesgo del encuentro más peligroso). Este problema se denomina *árbol generador con mínimo cuello de botella*. Todo árbol generador mínimo es también un árbol generador con mínimo cuello de botella, de modo que podemos aplicar los Algoritmos de Prim o Kruskal para resolver este problema.

El objetivo del trabajo práctico es implementar una aplicación visual para esta problemática. La aplicación debe contar con interfaces para realizar las siguientes acciones:

1. Cargar los nombres de los espías y definir los pares de espías que se pueden encontrar junto con la probabilidad de interceptación de ese encuentro (es decir, cargar el grafo G y la función de peso p de las aristas).
2. Resolver el problema de árbol generador con mínimo cuello de botella sobre los datos cargados, usando el Algoritmo de Prim o el Algoritmo de Kruskal.
3. Mostrar los resultados de este algoritmo, informando qué pares de espías se deben encontrar y cuál es la probabilidad de interceptación de cada encuentro.

Como objetivos adicionales no obligatorios, se pueden considerar los siguientes puntos:

1. Implementar tanto el Algoritmo de Prim como el Algoritmo de Kruskal y comparar sus tiempos de ejecución.
2. Leer los datos de los espías desde un archivo.
3. Una vez obtenido el árbol generador, recorrerlo con BFS para determinar en qué orden se deben realizar los encuentros.

creo que debería ir anotando el orden en que se va formando el arbol

Condiciones de entrega: El trabajo práctico se debe entregar por mail a los docentes de la materia. Además del código, se debe incluir un documento en el que se describa la implementación y se detallen las decisiones tomadas durante el desarrollo. Todas las clases de negocio deben tener tests unitarios. El trabajo práctico se puede hacer en grupos de tres o de cuatro personas.

Fecha de entrega: Martes 15 de octubre.