Proyecto 2

Árbol de expansión mínimo

Descripción

Programar los algoritmos de Prim y Kruskal para obtener el MST.

Instrucciones

- Implementar los siguientes algoritmos en Java:
  - Prim usando Heap de nodos
  - Kruskal usando DFS para ciclos
  - Kruskal usando Union-Find para ciclos.
- Probar los 4 algoritmos con el archivo anexo:
  - El archivo describe un grafo no dirigido con costos enteros de sus aristas (tomado de la referencia [1]).
  - Su formato es:
    - [Número de nodos] [Número de aristas]
    - [Nodo 1 de la arista 1] [Nodo 2 de la arista 1] [costo de la arista 1]
    - [Nodo 1 de la arista 2] [Nodo 2 de la arista 2] [costo de la arista 2]
    - …
  - Puede haber datos iguales y negativos.
- Reportar:
  - Solución Final: Costo total
  - Tiempo de ejecución
- Comente sus resultados.

API

Se requieren tres funciones:

Cada función lee el grafo del archivo que recibe como parámetro, obtiene su MST con el algoritmo dado y regresa su costo. Además imprime el tiempo de ejecución y las aristas del MST en el siguiente formato:

(nodoInicial, nodoFinal, costo)

```
// Prim
float prim(string archivo);
// Kruskal con DFS para detección de ciclos
float kruskalDFS(string archivo);
// Kruskal con Union-Find para detección de ciclos
float kruskalUF(string archivo);
```

Documentación

- Para este proyecto se requiere la documentación normal:
    - Manual del usuario con impresiones de pantalla
    - Descripción Técnica como comentarios en el código
    - Solución Final y análisis
    - Referencias (en caso de existir)

Referencias

[1] Tim Roughgarden. Algorithms- Design and Analysis- Part 1. Stanford-Coursera (2015). https://www.coursera.org/learn/algorithm-design-analysis.