

Informe proyecto 3: Palomas mensajeras

Integrantes: Tatiana Cesar, Jacqueline Dendrak y Sofia Méndez.

Enfoque de resolución

Lectura y modelado del grafo

El proyecto parte de la lectura del archivo aldeas.txt, que contiene la descripción del mapa de aldeas y las distancias que las separan.

Cada línea puede incluir dos aldeas conectadas por una arista ponderada (distancia en leguas) o una aldea aislada.

A partir de esta información, el módulo grafo.py construye un grafo no dirigido, almacenando sus nodos, aristas y pesos en estructuras adecuadas para su posterior procesamiento.

Elección del algoritmo

El objetivo del trabajo es determinar la forma más eficiente de conectar todas las aldeas minimizando la suma total de distancias.

Para lograrlo, se utiliza el algoritmo de Kruskal, el cual genera un Árbol de Expansión Mínima (MST).

Este enfoque permite conectar todas las aldeas con el menor costo global posible, independientemente del punto de partida, garantizando que no se formen ciclos y que todas las conexiones sean necesarias.

Implementación del algoritmo

El algoritmo de Kruskal se implementa en el módulo mst.py.

Primero se ordenan todas las aristas por peso y luego se van incorporando al árbol solo aquellas que no generen ciclos, utilizando la estructura Union-Find para gestionar las componentes conectadas.

Además, el mismo módulo incluye una función auxiliar para orientar el árbol desde una raíz (por defecto, la aldea "Peligros"), simulando el flujo de información a través de la red de aldeas.

Construcción del árbol orientado

Una vez obtenido el árbol mínimo, se define una dirección de transmisión:

cada aldea recibe la noticia desde una aldea vecina (su "padre") y la transmite a otras (sus "hijos").

De esta forma se puede determinar el recorrido de la información y el costo total de las distancias recorridas dentro de la red.

Estructura del proyecto

```
proyecto_3/
├── data/
│   └── aldeas.txt
├── docs/
│   └── README.md
├── modules/
│   ├── __init__.py
│   ├── grafo.py
│   └── mst.py
└── main.py
```

[grafo.py](#): Define la clase Graph, responsable de la lectura del archivo de datos y del modelado del grafo. También contiene la estructura UnionFind, utilizada para el algoritmo de Kruskal.

[mst.py](#): Contiene la implementación del algoritmo de Kruskal y la función orient_tree_from_root() que permite orientar el árbol mínimo desde una aldea raíz.

[main.py](#): Ejecuta el flujo general del programa: lectura del archivo, construcción del grafo, obtención del MST y orientación desde “Peligros”. Finalmente, imprime los resultados en consola.

aldeas.txt: Archivo de entrada con la descripción de las aldeas y sus distancias (en leguas).

docs/[README.md](#): Se muestra lo solicitado en el apartado entregables del propio ejercicio.

Resultados del programa

Al ejecutar [main.py](#) el programa muestra en la consola:

- Una lista alfabética de todas las aldeas
- Para cada aldea, quien le transmite la noticia y a quien se la transmite.
- La distancia total mínima en leguas que deben recorrer los mensajes.