

## Informe del proyecto 3: Palomas mensajeras

Integrantes: Ignacio De Chiara, Chiara Groglio Kremer, Carolina Pérez

### 1. Enfoque de resolución

#### 1.1 Lectura y modelado del grafo

Utilizamos una función **leer\_grafo(path\_archivo)** que interpreta el archivo aldeas.txt, el cual puede contener líneas con dos aldeas y una distancia entre ellas (arista bidireccional con peso) o una sola aldea sin conexiones (nodo aislado). A partir de esa lectura, se construye un diccionario que representa el grafo, y un conjunto con todas las aldeas involucradas.

#### 1.2 Elección del algoritmo

Inicialmente se implementó el algoritmo de Dijkstra, pensando en minimizar los caminos desde la aldea origen ("Peligros") hacia todas las demás. Sin embargo, luego de la devolución de los profesores, comprendimos que el problema requería una minimización global del costo de conectar todas las aldeas, independientemente del camino más corto desde un origen.

Por esa razón, se reemplazó Dijkstra por el algoritmo de Prim, que construye un árbol de expansión mínima. Este árbol garantiza que todas las aldeas están conectadas con el menor costo total posible.

#### 1.3 Implementación del algoritmo de Prim

El algoritmo fue implementado con una cola de prioridad (implementada por nosotros) para seleccionar siempre la arista de menor peso disponible. Se utiliza un diccionario predecesor para reconstruir el árbol y obtener qué nodo conecta con cuál, y un conjunto visitados para evitar ciclos.

La función `prim_con_monticulo(grafo, origen)` devuelve:

- Las distancias parciales usadas en la construcción del MST.
- Un diccionario de predecesores para cada aldea (quién le transmite la noticia).

#### 1.4 Construcción del árbol

Con el diccionario de predecesores, se arma el árbol de transmisión, registrando quién le transmite a cada aldea y a quién le transmite cada aldea.

Finalmente, se calcula el total de distancia que recorren las palomas sumando los pesos de las aristas que forman el árbol.

### 2. Estructura del proyecto

- **\*\*grafo.py\*\***

El archivo `grafo.py` utiliza el algoritmo de Prim para encontrar los caminos más cortos desde "Peligros" a las demás aldeas y construye un árbol de rutas mínimas a partir de los predecesores calculados.

- **\*\*main.py\*\***

Este archivo solo se encarga de ejecutar el programa, usando las funciones del módulo anterior, y muestra los resultados.

- **\*\*aldeas.txt\*\***

El archivo aldeas.txt contiene la descripción del mapa de aldeas, con las conexiones entre ellas y sus distancias expresadas en leguas. Se usa como entrada para construir el grafo del problema.

- **\*\*Monticulo.py\*\***:

Implementación genérica de un Montículo Binario.

- **\*\*cola\_prioridad.py\*\***

La clase `ColaPrioridad` contiene una cola con prioridades mínimas basada en el montículo binario, donde cada elemento se encola según una función de prioridad personalizada.

### 3. Resultados

El programa imprime:

- Una lista alfabética de todas las aldeas.
- Para cada aldea, desde quién recibe la noticia y a quién se la transmite.
- La distancia total en leguas recorrida por las palomas.