**Práctica 3**

**Propuesta de solución:**

El enunciado, en resumidas cuentas, nos pide realizar un grafo dirigido, basado en la población mallorquina, donde cada vértice representaría una población y cada arista representaría un camino/carretera/autopista que conecta dos vértices (poblaciones).

Para conseguir esto utilizaremos la estructura de datos de grafo, que cuenta con vértices y aristas. Utilizaremos la implementación del grafo utilizando listas enlazadas.

Los vértices estarían ubicados en un array de vértices con un puntero a la lista enlazada que contiene todas las aristas, que no son más que punteros a otros vértices y un float que represente la distancia entre dos vértices.

9

10

9

10

En este diagrama, la ciudad 1 estaría conectada con la ciudad 2 mediante una carretera de 10 km, y la ciudad 2, con la 3 mediante una carretera de 9 km. Su implementación, utilizando como ejemplo la ciudad 2, podría ser algo del siguiente estilo:

Ciudad 3

Ciudad 1

9 X

10

Ciudad 1

Ciudad 2

Ciudad 3

Es decir, una ciudad es un valor en un array que contiene un puntero a la lista de aristas, las cuales incluyen la longitud de la arista (distancia entre ciudades), una variable que indique el valor a la otra ciudad y por último un puntero a la siguiente arista en la lista.

Ya que se usan las ciudades como índice del array, es necesario asignar a cada ciudad un valor numérico para poder utilizar éste como índice. Para ello, utilizaremos un hashing.

**Estructuras de datos utilizados:**

Grafo:

El grafo es la principal estructura de datos que requerimos para implementar el mapa. Está implementado mediante el método de listas enlazadas, como fue explicado en la anterior sección.

Hashing:

El Hashing se utiliza para asignar valores numéricos a las diferentes ciudades, como se menciona en la sección anterior. En este caso hemos utilizado un hashing abierto debido a la necesidad de poder eliminar ciudades del conjunto.

Names table:

Es un conjunto de arrays que utilizamos para poder obtener el nombre de las ciudades mediante su valor de hashing.

Cola:

¿…?