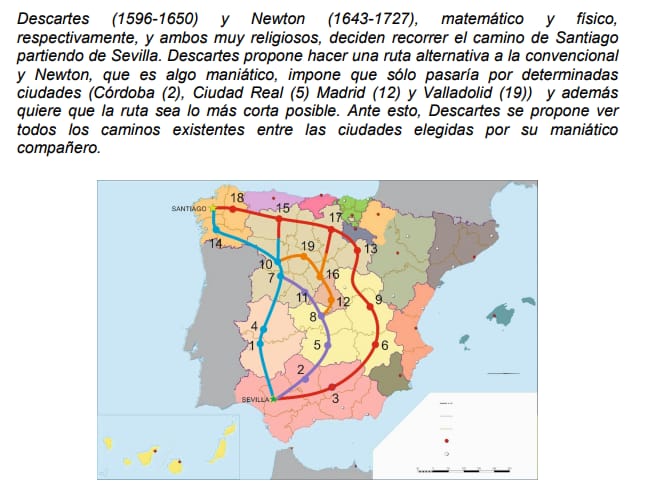
**Enunciado**

Descartes (1596-1650) y Newton (1643-1727), matemático y físico, respectivamente, y ambos muy religiosos, deciden recorrer el camino de Santiago partiendo de Sevilla. Descartes propone hacer una ruta alternativa a la convencional y Newton, que es algo maniático, impone que sólo pasaría por determinadas ciudades *( Córdoba (2) , Ciudad Real (5), Madrid(12) y Valladolid (19) )* y además quiere que la ruta sea lo más corta posible. Ante esto, Descartes se propone ver todos los caminos existentes entre las ciudades elegidas por su maniático compañero.

A usted, como estudiante del curso de Algoritmos de Estructuras de Datos, se le ha pedido realizar un programa que permita identificar los diferentes recorridos que pueden hacer Descartes y Newton en su viaje, pasando por determinadas ciudades, para este problema es necesario que sea modelado utilizando Grafos para la solución del problema. Su programa debe estar en la condición de solucionar el problema con dos versiones de algoritmos de grafos que se estudiaron en el curso. Además, su programa debe incluir una interfaz gráfica que permita utilizar las funcionalidades que responden a los requerimientos del problema.



**Identificación del problema**

Durante un viaje a España, los matemáticos Newton y Descartes quienes eran muy religiosos deciden realizar el [camino de Santiago](https://en.wikipedia.org/wiki/Camino_de_Santiago), pero para ello Descartes propone diferentes rutas. Sin embargo, Newton insiste en pasar solo por determinadas ciudades, y además desea que la ruta a recorrer sea la más corta posible. Entonces se requiere una ruta entre Sevilla y Santiago, pasando obligatoriamente por las ciudades de Córdoba, CIudad Real, Madrid y Valladolid.

Requerimientos Funcionales

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Definir el tipo de algoritmo de grafo |
| **Resumen:** | El usuario deberá determinar con que algoritmo de grafo quiere solucionar el problema |
| Entradas: | Algoritmo de grafo |
|  |  |
| Resultados: | Solución del problema con el algoritmo dado |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Permitir dar el recorrido más corto |
| **Resumen:** | Permitir definir el recorrido más corto para llegar al destino |
| Entradas: | Destino y origen |
|  |  |
| Resultados: | Recorrido más corto |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Permitir dibujar el grafo |
| **Resumen:** | Permitir dibujar el grafo del recorrido más corto para el usuario. |
| Entradas: | Ninguno |
|  |  |
| Resultados: | Grafo de recorrido |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #4. Determinar el tiempo de ejecución |
| **Resumen:** | Determinar cuanto se demora en hacer los recorridos cada algoritmo de grafo. |
| Entradas: | Ninguno |
|  |  |
| Resultados: | Tiempo de ejecución |
|  |  |

**Recopilación de información**

Se puede formular este problema como un grafo donde cada nodo representa una ciudad y cada camino o vértice representa la unión (camino entre una ciudad a otra) y el peso de cada uno de los caminos representa la distancia entre la ciudad A hacia la ciudad B.

Puesto que el problema sólo presenta las ciudades de España como números, es necesario conocer el nombre de cada ciudad a la cual le corresponde cada número. Después de investigar en el buscador Google, tenemos los siguientes datos:

0 Sevilla

1 Badajoz

2 Cordoba

3 Jaen

4 Cáceres

5 Ciudad Real

6 Albacete

7 Salamanca

8 Toledo

9 Cuenca

10 Zamora

11 Ávila

12 Madrid

13 Zaragoza

14 Ourense

15 León

16 Segovia

17 Burgos

18 Lugo

19 Valladolid

20 Santiago

Ya conocidos estos datos, para que el problema sea lo más acertado posible y cercano a la realidad, se necesita conocer ahora las distancia (en kilómetros) entre cada para de ciudades que se presenta en el problema. Posterior a una búsqueda en Google Maps se tiene lo siguiente:

* Sevilla-Badajoz 211.4 km
* Sevilla-Córdoba 133.3 km
* Sevilla-Jaén 228 km
* Badajoz-Cáceres 90 km
* Córdoba-Ciudad Real 188.3 km
* Jaén-Albacete 254.1 km
* Cáceres-Salamanca 206.9 km
* Ciudad Real-Toledo 112.9 km
* Albacete-Cuenca 140.2 km
* Salamanca-Zamora 62.6 km
* Salamanca-Ávila 100.4 km
* Toledo-Ávila 128.9 km
* Toledo-Madrid 74.4 km
* Cuenca-Zaragoza 261.1 km
* Zamora-Ourense 253.1 km
* Zamora-Valladolid 96.3 km
* Zamora-León 130 km
* Madrid-Segovia 86.2 km
* Zaragoza-Burgos 287.9 km
* Ourense-Santiago 100.8 km
* León-Lugo 212.7 km
* León-Burgos 173.3 km
* Segovia-Burgos 178.3 km
* Segovia-Valladolid 103.4 km
* Lugo-Santiago 95 km

**Búsqueda de soluciones creativas**

* **Algoritmos BFS y DFS**: Es posible buscar todas las combinaciones posibles mediante los algoritmos de BFS y DFS para encontrar todos los posibles caminos del grafo.
* **Algoritmos Prim, Kruskal o Dijkstra**: Dado que se desea, conocer la distancia mínima entre cada una de las ciudades, podemos aplicar el algoritmo de Dijkstra para encontrar el camino mínimo, o el Algoritmo de Prim o Kruskal, para el árbol de expansión mínimo.
* **Algoritmos de Floyd-Warshall:** para encontrar el camino mínimo que pase por los nodos requeridos

**Diseños preliminares**

Dado que los algoritmos de BFS y DFS representan algoritmos de fuerza bruta y para este caso son excesivamente ineficientes por ello se descartan.

Es por eso que lo más eficiente es utilizar los algoritmos de Dijkstra o Floyd-Warshall y Kruskal

**Selección de la mejor solución**

La mejor solución para resolver el problema son los algoritmos de Floyd Warshall, que posee la capacidad de retornar una matriz de pesos de todos los caminos posibles y en el algoritmo de Prim es posible hacer un árbol de expansión mínima escogiendo un nodo inicial. Es por eso que el algoritmo de Kruskal, que posee un orden, no es suficiente para resolver el problema.