



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



ALGORÍTMICA

Grado en Ingeniería Informática
ETSIIT

Prácticas de laboratorio

Greedy

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA
ARTIFICIAL
UNIVERSIDAD DE GRANADA**



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



1. Objetivos

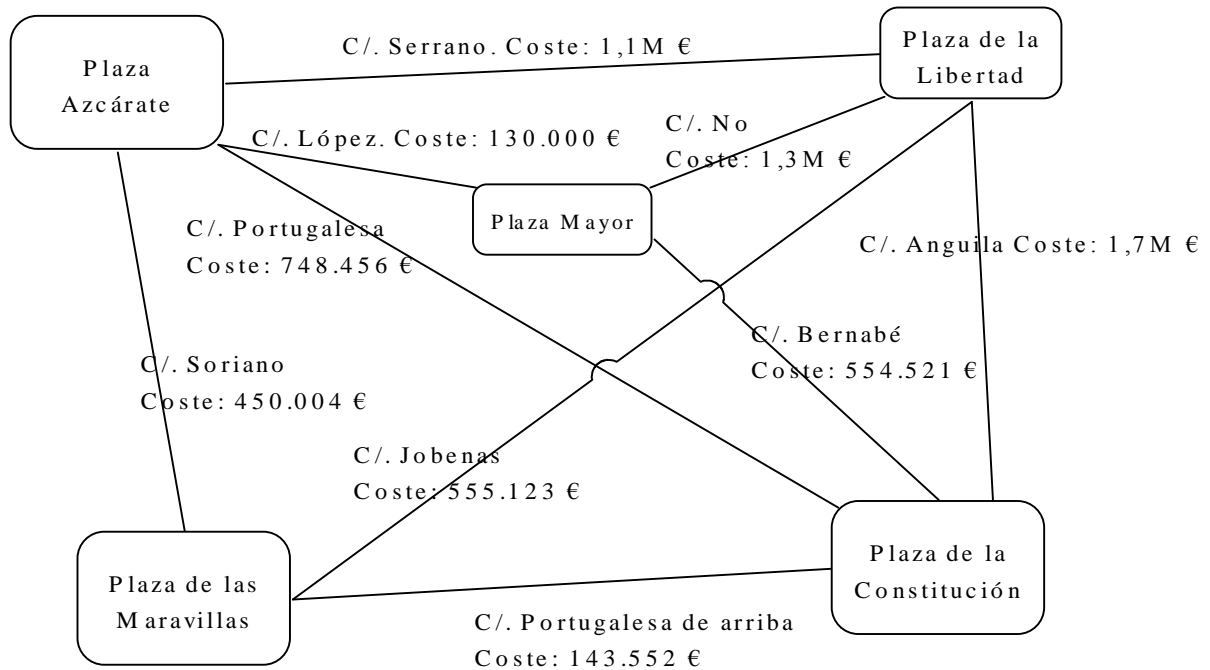
El objetivo de esta práctica es desarrollar las habilidades para analizar problemas resolubles mediante la técnica de diseño de algoritmos voraz. Para ello, se presentará un problema que se analizará y resolverá utilizando esta técnica durante las horas de prácticas en laboratorio.

2. Enunciado del problema

El alcalde de “Algovilla del Tuerto”, un conocido pueblo, desea renovar el embaldosado de las calles de su localidad. Sin embargo, las arcas del ayuntamiento no están muy saneadas y no se puede permitir embaldosar todas las calles del pueblo. El encanto de Algovilla reside en sus múltiples plazas (una en cada intersección de calles), que son monumentos locales, y en la belleza de pasear entre dichas plazas en época de verano, por lo que es un atractivo turístico que trae riqueza y trabajo a la localidad en esta época del año. El no realizar el embaldosado puede disminuir el atractivo del pueblo, la visita turística y, por tanto, reducir el desahogo económico que se produce en verano para sus habitantes, debido al turismo.

Para solucionar el problema, el concejal de urbanismo ha propuesto la siguiente solución: Asfaltar el mínimo número de calles, siempre que se pueda llegar desde una plaza a cualquier otra a través de calles asfaltadas. Así, cualquier turista podrá disfrutar de los paseos entre estos monumentos. Como asesor, se te requiere que formules el problema y lo resuelvas, proporcionando una solución que permita viajar desde cualquier plaza a cualquier otra plaza, con la restricción de que el paseo se realice siempre por una calle asfaltada y que el coste de asfaltar las calles necesarias para ello sea mínimo. Se te proporcionará información sobre qué plazas están unidas entre sí directamente por una única calle y el coste de asfaltarlas (suponer coste igual a $+\infty$ cuando no exista una calle que une dos plazas de forma directa).

Un ejemplo serían las siguientes plazas, calles y costes:



3. Estructuras de Datos

En este apartado se pide **diseñar e implementar** las estructuras de datos necesarias para representar:

- La lista de los nombres de plazas existentes en el problema.
- La lista de calles que unen cada par de plazas.
- El coste de asfaltar cada calle.
- Las distancias que hay entre las ciudades (asumir distancia igual a $+\infty$ para dos ciudades que no tengan conexión directa).
- El conjunto de calles que deben ser asfaltadas con mínimo coste, junto con el valor de dicho coste, de modo que se pueda viajar desde cualquier plaza a otra utilizando sólo calles asfaltadas.

Adicionalmente, se deberá implementar métodos para:

- Leer desde fichero de entrada una instancia del problema: lista de plazas, calles que unen cada plaza y el coste de asfaltar cada calle. Para aquellas calles que no existan para unir dos plazas de forma directa se utilizará un valor de coste igual a $+\infty$.
- Guardar a fichero de salida el conjunto de calles que deben ser asfaltadas, junto con el coste total de la obra.



Estas componentes serán discutidas en horario de clase de laboratorio, planteadas por el profesor fomentando el debate entre los alumnos, y deberán ser resueltas e implementadas también en horario de clase.

4. Diseño del algoritmo voraz

En este apartado se deberán analizar las siguientes cuestiones de diseño:

- Analizar y describir las componentes de diseño greedy, indicando cuál es el cometido de cada una en un diseño voraz.
- Diseñar cada componente greedy para solucionar el problema.
- Adaptar el esquema general greedy para solucionar el problema.
- Justificar el diseño realizado y explicar, mediante un ejemplo simple, cómo sería el funcionamiento del algoritmo diseñado.
- Demostrar si el algoritmo devuelve la solución óptima al problema o, en caso de no ser posible, exponer un contraejemplo que ponga de manifiesto la no optimalidad.
- Indicar el orden de eficiencia del algoritmo (en el caso peor), indicando las variables de las que depende el tamaño de caso del problema.
- Proporcionar una implementación del algoritmo.

Cada una de estas actividades será discutida conjuntamente entre alumnos y profesor en horario de prácticas de laboratorio, y deberán ser resueltas e implementadas también en horario de clase.

5. Implementación

En este apartado, se pide implementar el algoritmo dentro de un programa que realice lo siguiente:

- Lea desde fichero (pasado por línea de comandos) una lista de plazas, calles y costes para asfaltar las calles. El formato de este fichero será discutido en clase y acordado entre alumnos y profesor.
- Calcule las calles a asfaltar y el coste total, y guarde estos resultados a fichero de salida (pasado también al programa por línea de comandos).

6. Tareas a realizar

Se deben analizar, diseñar e implementar un algoritmo voraz que resuelva el problema planteado en el apartado 2 de este guión. Concretamente:



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



- a. Para el apartado 3, diseñar las estructuras de datos básicas necesarias que servirán para representar el problema y la solución al mismo.
- b. Para el apartado 4, diseñar las componentes greedy de un algoritmo para resolver el problema, y adaptar el esquema general greedy a estas componentes para solucionarlo. Adicionalmente, en caso de proporcionar solución óptima, se deberá incluir demostración de optimalidad o, en su caso, contraejemplo que ilustre lo contrario.
- c. Para el apartado 5, código fuente documentado del programa y algoritmo de resolución del problema.