



UNIVERSIDAD  
DE GRANADA

*Este documento está protegido por la Ley de Propiedad Intelectual ([Real Decreto Ley 1/1996 de 12 de abril](#)).  
Queda expresamente prohibido su uso o distribución sin autorización del autor.*

# Algorítmica

## 2º Grado en Ingeniería Informática

### Guión de prácticas

#### Algoritmos Voraces

1. Objetivo.....	2
2. Diseño de algoritmos voraces.....	2
3. Ejercicios propuestos.....	3
4. Evaluación de la práctica.....	5
5. Entrega y presentación de la práctica.....	5

© Prof. Manuel Pegalajar Cuéllar  
Dpto. Ciencias de la Computación e I. A.  
Universidad de Granada



DECSAI

**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

# Algoritmos Voraces

## 1. Objetivo

El objetivo de la práctica consiste en que el alumno sea capaz de analizar un problema y resolverlo mediante la técnica Greedy (algoritmos voraces), siendo capaz de justificar su eficacia en términos de optimalidad. Se expondrá un problema que será resuelto en clase por el profesor. Posteriormente, se expone un conjunto de problemas que deberán ser resueltos por el estudiante.

## 2. Diseño de algoritmos voraces

### 2.1. Descripción del problema de ejemplo

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 que se muestran en la Figura 1:

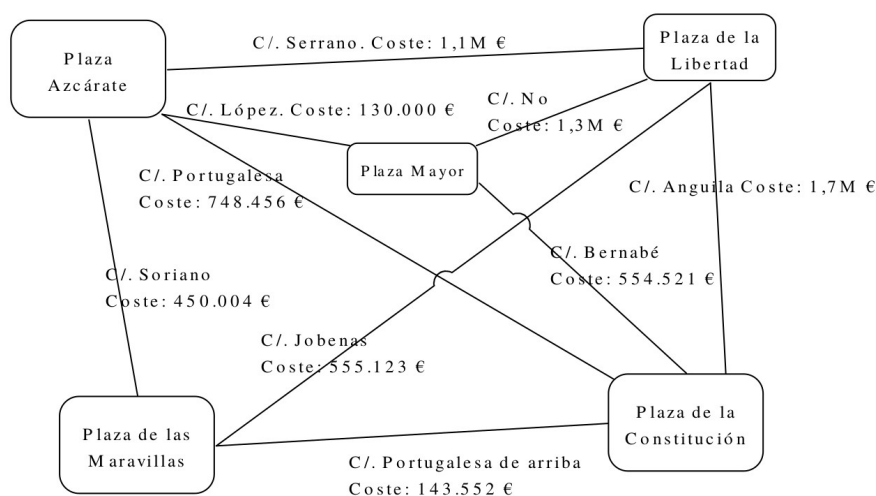


Figura 1: Esquema con las plazas del pueblo, las calles que las unen y el precio de asfaltar cada calle.

## 2.2. Solución: Diseño de componentes

El problema lo modelamos como un grafo, donde cada vértice es el nombre de una plaza y cada arista una calle que une dos plazas. El grafo resultante cumple con las propiedades de ser no dirigido, conexo y ponderado con pesos no negativos. El objetivo consiste en encontrar un subconjunto de aristas que unan todos los vértices del grafo y cuya suma de pesos sea mínima. Por tanto, nos encontramos con que puede ser resuelto como instancia del Árbol Generador Minimal. Sus componentes de diseño serían:

- Lista de candidatos: Las calles que podrían asfaltarse.
- Lista de candidatos usados: Las calles que han sido consideradas previamente para insertarse en la solución, haya sido factible su inserción o no.
- Criterio de selección: Se seleccionará la calle de coste mínimo.
- Criterio de factibilidad: Una calle se insertará en la solución si, al insertarla, no produce ciclos dentro del grafo solución.
- Función solución: Una subsolución será solución final al problema cuando las calles seleccionadas unan todas las plazas. Es decir, si hay  $N$  plazas, una subsolución se considerará solución si contiene  $N-1$  calles.
- Función objetivo: Minimizar la suma de los pesos de las aristas que componen la solución al problema.

Con este diseño de componentes, podemos pasar a adaptar la plantilla Greedy para resolver el problema de la siguiente forma:

**Algoritmo  $S=Greedy(G=<V, A>$ : Grafo de entrada)**

```
S={ } # Conjunto vacío
C= A
Mientras  $|S| < |V| - 1$ , hacer:
    x= Seleccionar arista más corta en C
    C= C \ {x}
    Si S U {x} no forma ciclos en V, hacer:
        S= S U {x}
Devolver S
```

El algoritmo diseñado es óptimo (ver apuntes del problema A.G.M. de teoría).

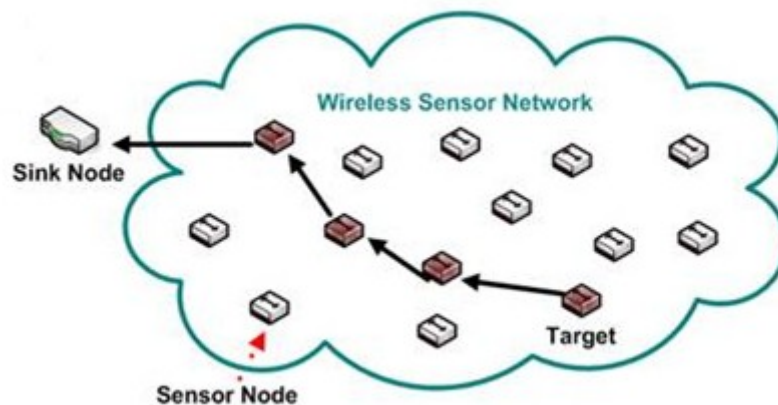
## 3. Ejercicios propuestos

Para cada uno de los ejercicios siguientes, se debe:

- Diseñar e implementar un algoritmo Greedy que resuelva el problema.
- Analizar la optimalidad (mediante demostración o contraejemplo) del algoritmo diseñado.
- Proponer un ejemplo de uso (instancia del problema) representativo, y explicar cómo funciona el algoritmo sobre este ejemplo, paso a paso.

### 3.1. Ejercicio 1

Una red de sensores inalámbrica está compuesta por múltiples nodos sensores desplegados en un entorno (invernadero, campo de cultivo, instalación industrial, perímetro de vigilancia de seguridad, ...), cada uno equipado con un transmisor de datos inalámbrico de un alcance reducido. Cada cierto tiempo, cada sensor debe enviar los datos recolectados del entorno a un servidor de datos central (sink). Sin embargo, la distancia entre el nodo sensor y el servidor central puede ser elevada como para enviar todos los datos directamente, por lo que será necesario, en ocasiones, enviar los datos por otros nodos sensores que hagan de enlace intermedio (ver Figura 1). Nos interesa enviar los datos con la máxima velocidad posible por lo que, para cada par de nodos sensores de la red  $n_i, n_j$  (entre los que se incluye el servidor central), conocemos el tiempo de envío entre ambos nodos como  $t(n_i, n_j)$  – el tiempo que se tarda en enviar los datos desde el nodo  $n_i$  al nodo  $n_j$  -. El valor  $t(n_i, n_j)$  podría tener valor infinito si la red inalámbrica no permite enviar datos directamente desde el nodo  $n_i$  hasta el nodo  $n_j$ . Se pide: desarrollar un algoritmo que nos permita conocer por cuáles nodos sensores intermedios debe enviar los datos cada nodo sensor, hasta llegar el servidor central, de modo que se tarde el mínimo tiempo en la transmisión desde cada nodo hasta el servidor central.



*Figura 2: Configuración de envío desde un sensor inalámbrico (target) al receptor (sink). Fuente: <https://elb105.com/pfc-design-and-implementation-of-a-dynamic-wireless-sensor-network/>*

## 3.2. Ejercicio 2

Un autobús realiza una ruta determinada entre su origen y su destino ( $n$  kilómetros en total). Con el tanque de gasolina lleno, el autobús puede recorrer  $k$  kilómetros sin parar. El conductor dispone de un listado con las gasolineras existentes en su camino, y el punto kilométrico donde se encuentran. Se pide: Diseñar un algoritmo greedy que determine en qué gasolineras tiene que repostar el conductor para realizar el mínimo número de paradas posible.

## 3.3. Ejercicio 3

Se tiene un buque mercante cuya capacidad de carga es de  $k$  toneladas, y un conjunto de contenedores  $c_1, \dots, c_n$  cuyos pesos respectivos son  $p_1, \dots, p_n$  (expresados también en toneladas). Teniendo en cuenta que la capacidad del buque podría ser menor que la suma total de los pesos de los contenedores, se pide: Diseñar un algoritmo que permita decidir qué contenedores hay que cargar para maximizar la suma de los pesos de los contenedores a transportar en el barco.

## 4. Evaluación de la práctica

Se deben resolver todos los ejercicios propuestos en el apartado 3 de este guión. Cada ejercicio se valorará sobre 10 de la siguiente forma:

1. **(3 puntos)** Diseño de componentes
2. **(3 puntos)** Diseño del algoritmo
3. **(1 punto)** Estudio de optimalidad (demostración o contraejemplo)
4. **(2 puntos)** Ejemplo paso a paso de la explicación del funcionamiento del algoritmo para una instancia pequeña propuesta por el estudiante.
5. **(1 punto)** Correcto funcionamiento de la implementación.

La valoración de la práctica se dará como una calificación numérica entre 0 y 10. **Todos los problemas propuestos tendrán la misma calificación máxima de 10/3.**

## 5. Entrega y presentación de la práctica

Se deberá entregar un documento (memoria de prácticas) realizado en equipos de 3 personas, conteniendo los siguientes apartados:

1. Solución a los problemas propuestos, conteniendo los apartados descritos en el apartado 4 de este guión, para cada problema propuesto.

La práctica deberá ser entregada por PRADO, en la fecha y hora límite explicada en clase por el profesor. No se aceptarán, bajo ningún concepto, prácticas entregadas con posterioridad a la fecha límite indicada. La entrega de PRADO permanecerá abierta con, al menos, una semana de antelación antes de la fecha límite, por lo que todo alumno tendrá tiempo suficiente para entregarla. **La práctica deberá ser enviada por todos los integrantes del equipo de trabajo.**

El profesor, en clase de prácticas, realizará controles de las prácticas a discreción, que consistirán en presentaciones de los estudiantes de cada equipo (powerpoint) y/o entrevistas individuales con el fin de asegurar de que los estudiantes alcanzan las competencias deseadas. Estas entrevistas y/o presentaciones se realizarán en las sesiones de evaluación de prácticas, previamente anunciadas en clase por el profesor.

La **no asistencia** a una sesión de evaluación de prácticas por un estudiante supondrá la **calificación de 0 (no presentado) a la práctica que deba presentar, independientemente de la calificación obtenida en la memoria de prácticas.**

La presentación/entrevista de cada práctica podrá modificar la calificación final en un rango de calificaciones [-1, +1] con respecto a la calificación obtenida en la memoria de prácticas.

**IMPORTANTE: Antes de las sesiones de evaluación, cada estudiante deberá prepararse para:**

- Realizar una presentación powerpoint de la práctica de **máximo 5 minutos**, si el profesor lo exige.
- Conocer a fondo todos los algoritmos resueltos, así como los pasos para diseñar e implementar los algoritmos.
- Conocer las respuestas a las preguntas requeridas en el apartado 4 para cada problema.

El desconocimiento o la ausencia del material indicado podrá suponer la calificación de 0 en la práctica.