



**UNIVERSIDAD  
DE GRANADA**

**E.T.S. DE INGENIERÍAS INFORMÁTICA y DE  
TELECOMUNICACIÓN**

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

**Algorítmica**

**Guión de Prácticas**

**Práctica 3: Algoritmos Voraces (Greedy)**

Curso 2023-2024

Grado en Informática

# Índice

1. Enunciado de los problemas	2
2. Trabajo a realizar	3
2.1. Tareas . . . . .	3
3. Memoria	4
4. Evaluación de la práctica	4

# Objetivo

El objetivo de esta práctica es que el estudiante comprenda y asimile el funcionamiento de la técnica de diseño de algoritmos voraces. Esta práctica será realizada por grupos de alumnos. Cada grupo deberá resolver los problemas planteados adaptando los algoritmos a la estructura voraz vista en teoría. Para estos problemas es necesario encontrar la solución exacta del mismo. Además, tendrá que abordar el diseño de heurísticas, algoritmos aproximativos, basados en la filosofía de los métodos voraces para encontrar soluciones a un problema muy complejo, para el que no se conocen algoritmos eficientes. En la sección que sigue se muestran los enunciados de los problemas a considerar.

## 1. Enunciado de los problemas

En esta sección se incluyen los enunciados de los problemas que tienen que resolver cada grupo de alumnos. Los problemas considerados son los cinco primeros que se describen esta sección. Para todos ellos hay que diseñar un algoritmo que dé una solución válida, es decir, que cumpla los requisitos y que dé el óptimo cuando sea el caso. Sin embargo, el último problema, denominado "Viajante de comercio" es un problema especialmente complejo<sup>1</sup> y se sospecha que no existen algoritmos que den soluciones optimales para todos los casos en un tiempo "aceptable". En ese caso, como solución aceptable se diseñan algoritmos que dan soluciones aproximadas, heurísticas. Este último problema habrá de ser abordado por todos los grupos de trabajo, además, del problema concreto que se les asigne.

### P1: Hijo predilecto

Un individuo desea repartir sus  $N$  bienes entre sus dos hijos. Para cada bien se conoce el valor (positivo) del mismo. Previo al fallecimiento del individuo, el juzgado, de una forma ciega, determina un valor  $k$ , que nos indica el número de bienes que se le deben asignar a un hijo (al otro se le asignan  $N - k$ ). Conocido estos datos, el individuo desea distribuir los bienes entre sus hijos de forma que uno de ellos salga lo más beneficiado posible. El beneficio obtenido por cada hijo se define como la suma de los beneficios de los bienes que se les lega.

### P2: Asignación de exámenes a aulas

La Escuela Técnica Superior de Ingeniería Informática y de Telecomunicaciones (ETSIIT) está planificando los exámenes finales para las distintas asignaturas. Se deben programar  $N$  exámenes en un día determinado, y para cada examen, el profesorado ha especificado el tiempo requerido para su realización y la hora de inicio ideal.

El centro cuenta con  $m$  aulas disponibles, donde  $m$  es mayor que  $N$ , lo que garantiza la viabilidad de programar los exámenes. Sin embargo, cada vez que se utiliza un aula, el centro debe asegurar su vigilancia, lo que implica contratar a un vigilante distinto para cada aula utilizada.

Se solicita diseñar un algoritmo que, de manera eficiente en términos de tiempo, permita garantizar que los exámenes se realicen con el menor costo posible para la escuela.

### P3: Problema de camino mínimo

Tenemos un conjunto de  $n$  ciudades (puntos en un plano), cada una definida por las coordenadas en el mapa  $(x_i, y_i)$ , con  $i = 1, \dots, n$ . Cada ciudad se puede conectar por carretera con un subconjunto de ciudades, asegurando que la red de vías que se genere crea un grafo conexo.

La distancia entre dos ciudades viene dada por la distancia euclídea entre sus coordenadas.

$$\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Dada una ciudad de origen,  $x$ , y una ciudad de destino,  $y$ , se pide encontrar el camino de costo mínimo que permita viajar de  $x$  a  $y$ .

### P4: Problema del viajante de comercio

Tenemos un conjunto de  $n$  ciudades (puntos en un plano), cada una definida por las coordenadas en el mapa  $(x_i, y_i)$ , con  $i = 1, \dots, n$ . La distancia entre dos ciudades viene dada por la distancia euclídea entre sus coordenadas.

$$\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

El problema de viajante de comercio consiste en encontrar el orden en el que un viajante, partiendo de la ciudad de origen (por ejemplo  $(x_1, y_1)$ ) pase por todas y cada una de las ciudades una única vez, para volver a la ciudad de partida, formando un ciclo.

El costo del ciclo será la suma de las distancias que hay entre las ciudades.

El problema original del viajante de comercio consiste en encontrar el ciclo de costo mínimo entre todas las posibilidades existentes.

Aunque este problema es NP-Difícil y por tanto no podemos esperar encontrar una solución óptima al mismo, lo que se pretende en esta práctica es utilizar la estrategia greedy para encontrar una solución aproximada que puede ser de utilidad en situaciones como las que se plantea en este problema.

## 2. Trabajo a realizar

Para alcanzar el objetivo previsto con esta práctica, los alumnos deberán realizar las tareas que se detallan en la sección siguiente, documentando el trabajo realizado en una memoria con el contenido y estructura que se indica en la sección [3](#).

### 2.1. Tareas

1. Estudiar en profundidad cada problema, asegurándose de comprender bien las entradas y salidas involucradas, así como las relaciones entre éstas. Deberán identificar el tipo de información que define cada caso del problema y las magnitudes que definen su tamaño.
2. Diseñar e implementar un generador de casos para el problema.
3. Diseñar un algoritmo voraz para resolver el problema.

4. Demostrar o justificar la validez del algoritmo, es decir, que la solución que proporciona es la correcta (óptima en el caso de problemas de optimización).
5. Implementar el algoritmo
6. Para el problema del *viajante de comercio*, habrá que diseñar tres métodos aproximados (heurísticas). El estudio de estos métodos incluirá su diseño, explicando cómo funcionan, su implementación y estudio de análisis de eficiencia.

### 3. Memoria

Todo el trabajo realizado debe redactarse en una memoria. La estructura de este documento será la siguiente:

1. Portada. Incluyendo las denominaciones de titulación, asignatura y práctica. También el nombre completo de los alumnos que forman el grupo y su dirección de correo electrónico.
2. Autores. Indicar el % del trabajo realizado por cada alumno, especificando qué tareas ha realizado cada uno.
3. Objetivos. Descripción del objetivo de la práctica.
4. Definición del problema. Descripción de los casos usados en la evaluación de la eficiencia. Descripción completa del entorno de análisis: hardware, sistema operativo, compilador, etc. empleados.
5. Algoritmo Voraz. Diseño y justificación de su validez. Detalles de implementación.
6. Para el problema del viajante de comercio, análisis comparativo de las heurísticas en términos de la calidad de las soluciones que construyen y tiempos de ejecución.
7. Conclusiones.

### 4. Evaluación de la práctica

Esta práctica se realizará por grupos. Los alumnos habrán de entregar dos archivos en la actividad correspondiente incluida en la página de la asignatura en la plataforma Prado.

- El primer archivo será la memoria, en formato pdf.
- El segundo será el código fuente de las implementaciones realizadas empaquetado en un fichero .zip. La Figura 1 ilustra la estructura que se debe crear donde para cada problema estudiado tendremos una carpeta distinta, llamadas P1, P2 y P3. En cada carpeta el código estará organizado en tres subcarpetas de nombres: Generador, Epecifico y Greedy. Cada una de esas carpetas incluirá todos los módulos de código fuente necesarios para generar el programa binario correspondiente las respectivas implementaciones de: generador de casos del problema, algoritmo específico y algoritmo diseñado.

Figura 1: Estructura para entregar en prácticas

```
P1
|- Generador
|- Greedy
|- Instancias
|  - ni1.txt
|  - ni2.txt
|  - ...
|- Makefile
P2
| ...
P3
| ...
```

El generador de casos del problema deberá almacenar las instancias que cree en el directorio Instancias. Dicho programa deberá aceptar como entradas al menos dos parámetros, el primero el tamaño de la instancia a crear y el segundo el nombre del fichero salida para dicha instancia. Se generará una instancia y el resultado se almacenará en el directorio instancias con en nombre indicado.

Los programas aceptarán argumentos en línea de órdenes, que identificarán el caso que trata de resolver (por ejemplo, el nombre de un fichero que incluirá los datos) y enviarán a la salida estándar la solución calculada.

Además, en la carpeta raíz habrá un único fichero Makefile cuyo objetivo por defecto construirá los tres binarios de nombres generador, especifico y greedy, respectivamente.

La fecha límite para entregar la memoria es el día 24 de abril de 2024 a las 23:59 horas. Además de hacer el trabajo y entregar la memoria, cada equipo tendrá que elaborar una breve presentación del trabajo realizado que expondrá públicamente en clase de prácticas, de acuerdo a la convocatoria establecida por el profesor. Es **obligatoria** la participación de todos los miembros del grupo en la exposición.