



# Práctica 1 - Métodos voraces

Créditos: 0.6 (15 horas); Peso: 10 %

Eres un programador *freelance* que ofrece servicios de optimización de procesos para empresas. Te han llegado dos ofertas de trabajo que no puedes rechazar. Uno es de la red de restaurantes **Sopas y Caldos S.L.**, especializada en sopas, caldos y estofados; y el otro es del **Ministerio de Transporte, Movilidad y Agenda Urbana** para desarrollar un plan de carreteras para conectar varias comarcas.

Las especificaciones de cada contrato son:

## 1. Problemas a resolver

### 1.1. Optimizar producción de sopas

Sopas y Caldos S.L. se plantea optimizar la producción de ollas de sopas, caldos y estofados para cada uno de sus restaurantes. Se pide:

- Dado el **presupuesto** total diario del restaurante y teniendo el índice de popularidad de cada plato (especificada en el registro total de comandas recibidas en la historia del restaurante) y el precio de producción de cada olla. Se pide optimizar que ollas de sopa (o partes de ella) se han de producir para optimizar la popularidad total de las sopas.

Por simplificación del problema, cada olla de sopa se hace una sola vez y se puede hacer una parte de olla (entre 0 y 1) siendo 0 que no se hace esa sopa y 1 que la olla se hace entera. El coste de cada sopa es proporcional a la cantidad que se hace.

### 1.2. Plan de carreteras

A partir de una red de localidades se te da el coste de hacer una carretera entre cada par. Crear una red que minimice el coste de construcción con dos condiciones:

1. No puede haber zonas aisladas, desde cada localidad se debe poder llegar a cualquier otra por toda la red de carreteras.
2. Hay que minimizar costes, por lo que no puede haber ciclos cerrados en la red. E.g si la ciudad 'A' está conectada a la ciudad 'B' y la ciudad 'B' a la ciudad 'C', no se ha de construir una carretera que conecte de la ciudad 'A' a la 'C'.

## 2. Informe

Junto con la solución planteada se pide que se respondan a las siguientes preguntas que se plantean en el *Notebook* de la práctica:

1. Diga la complejidad temporal y espacial de los algoritmos que se pidan.
2. Responda a las preguntas:
  - ¿Qué algoritmos voraces has utilizado?
  - ¿Qué estructuras de datos has utilizado?
  - Respecto a la planificación de las sopas (Justifica la respuesta):
    - ¿La solución es óptima (maximiza siempre el ratio de popularidad) o es aproximada (encuentra un máximo local)?
    - ¿Qué efectos a largo plazo tendría utilizar el criterio de popularidad para la planificación? Ten en cuenta que la popularidad se calcula como todas las veces que esa sopa se ha pedido en el pasado.

- Respecto a la red de carreteras (Justifica la respuesta):
  - ¿La solución es óptima (minimiza siempre el valor pedido) o es aproximada (encuentra un mínimo local)?
  - Se te da un coste por cada par de localidades, pero, si existiesen carreteras que no se puede hacer, ¿qué efectos tendría en tu solución?

### 3. Rúbrica

Si no cumple los requisitos (no pasa ninguno de los tests), se utilizan librerías no nativas de *Python* o no se contestan a las preguntas, se tendrá una puntuación de cero en toda la práctica.

No se pueden cambiar las cabeceras de las clases y funciones del *Notebook* provisto, pero sí se puede y se recomienda crear funciones y clases adicionales para facilitar el buen diseño de la solución.

#### 3.1. Código 6/10

- Código repetido, inútil o no general: -0.25 puntos por cada ocurrencia.

##### 3.1.1. Sopas y Caldos S.L. (50 %)

- Funcionalidad correcta (supera todos los tests): 7 puntos
- Complejidad temporal óptima: 2 puntos
- Complejidad espacial óptima: 1 punto

##### 3.1.2. Ministerio de Transporte, Movilidad y Agenda Urbana (50 %)

- Funcionalidad correcta (supera todos los tests): 6 puntos
- Complejidad temporal óptima: 1.5 puntos
- Complejidad espacial óptima: 2.5 puntos

#### 3.2. Informe 4/10

- Análisis de la complejidad temporal y espacial: 4 puntos
- Preguntas: 6 puntos

### 4. Entrega

La entrega de la práctica consistirá en un fichero `.ipynb` con el *Notebook* con el código y las preguntas respondidas.

Los ficheros deberán tener el siguiente nombre:

`<ApellidosPrimerAlumno>_<ApellidosSegundoAlumno>_voraz.ipynb`

Así, si los alumnos fueran “José Luis Garrido Labrador” e “Ismael Ramos Pérez”, la entrega sería:

- `GarridoLabrador_RamosPerez_voraz.ipynb`

**Nota importante 1:** si el documento no tiene este formato de nombre, la práctica entera tendrá una penalización de **dos puntos** sobre el total.

**Nota importante 2:** en caso de que se entregue por parejas **ambas personas** deberán hacer la entrega y deberán entregar **el mismo fichero**, si uno de la pareja no hace la entrega se evaluará como “No presentado”. En caso de que estos ficheros sean diferentes (bajo verificación por SHA256 y manual) se tendrá la penalización de **dos puntos** y se tendrá en cuenta cada práctica por separado.

**Nota importante 3:** en el caso de que se suba un fichero que no es un *Jupyter Notebook*, **esté corrupto** o **tenga fallos de sintaxis**, alguna parte del código, la práctica tendrá una nota de 0. Por tanto, verificar que la entrega ha sido correcta. Tampoco se permite en el código llamadas a funciones del sistema, de tener alguna de estas llamadas la práctica tendrá también una nota de 0.