

# Métodos algorítmicos en resolución de problemas II

Grado en Ingeniería Informática

Doble Grado en Ingeniería Informática y Matemáticas

CONVOCATORIA EXTRAORDINARIA DE JUNIO

Curso 2021/2022

---

## 1. Programación dinámica [2,75 puntos]

Dada una secuencia de palabras  $ps = p_1 \dots p_n$ , queremos encontrar la subsecuencia más larga de la secuencia que cumpla que cada palabra sea prefijo de la siguiente. Por ejemplo, si  $ps$  es la secuencia  $ca, p, d, pre, casa, de, prenda, precio, prendada, decaído$ , la solución sería  $p, pre, prenda, prendada$ , de longitud 4.

Se pide desarrollar un algoritmo de programación dinámica para resolver este problema suponiendo que se dispone de una función booleana  $prefijo(c_1, c_2)$  para saber si una cadena  $c_1$  es prefijo de otra  $c_2$ . Se valorarán todos los pasos: definición de la recurrencia, implementación del algoritmo y análisis de los costes en tiempo de ejecución y memoria adicional del mismo.

## 2. Ramificación y poda [2,75 puntos]

Dos hermanos, Agustín y Beatriz, han de ayudar en casa con  $n$  tareas domésticas. Cada tarea ha de ser realizada por uno y solamente uno de los dos hermanos. Ambos consideran que estas tareas son sumamente desagradables y a cada tarea  $i$  ( $1 \leq i \leq n$ ) le asignan un coste  $c_i > 0$ , que es igual para ambos. El coste de las tareas asignadas a una persona es la suma de los costes de dichas tareas. Se desea encontrar una distribución de las  $n$  tareas entre los hermanos que minimice el máximo entre el coste  $c_A$  de las tareas asignadas a Agustín y el coste  $c_B$  de las tareas asignadas a Beatriz.

Se pide diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema. Se valorarán todos los pasos: descripción del espacio de soluciones, definición y discusión de las cotas optimistas y pesimistas, implementación del algoritmo y análisis del coste en tiempo.

## 3. NP-completitud [1,5 puntos]

Dado un conjunto  $U$  de objetos, una familia  $S = \{S_1, \dots, S_n\}$  de subconjuntos de  $U$  y un número  $k$ , el problema SET-COVER consiste en determinar si existe una subfamilia  $T \subseteq S$  de a lo sumo  $k$  subconjuntos cuya unión es igual a  $U$ . Demostrar que el problema SET-COVER es NP-completo utilizando el problema de decisión de la cobertura de vértices PDCV.