

# Métodos Algorítmicos en Resolución de Problemas

Grado en Ingeniería Informática

Hoja de ejercicios 7

Curso 2018-2019

## EJERCICIOS DE RAMIFICACIÓN Y PODA

**Ejercicio 1** Sean  $n$  programas  $P_1, P_2, \dots, P_n$  que hay que almacenar en un disco. El programa  $P_i$  requiere  $s_i$  kilobytes de espacio de disco, y la capacidad del disco es  $D$  kilobytes, donde  $D < \sum_{i=1}^n s_i$ . Desarrollar un algoritmo que elija el subconjunto de los programas que tienen que ser almacenados de forma que se maximice la capacidad de disco utilizado.

**Ejercicio 2** Se tiene un sistema monetario formado por un conjunto finito  $M = \{m_1, m_2, \dots, m_n\}$  de tipos de monedas. Suponiendo que la cantidad disponible de monedas del tipo  $m_i$  es  $c_i$ , se quiere pagar una cantidad  $C$  utilizando un número total de monedas mínimo. Desarrollar un algoritmo que devuelva la mejor solución.

**Ejercicio 3** Se tienen matrices de números naturales  $M[1..n, 1..n]$  y  $H[1..n, 1..n]$  tales que  $M[i, j]$  indica la preferencia de la mujer  $i$  por el hombre  $j$ , y  $H[i, j]$  la preferencia del hombre  $i$  por la mujer  $j$ , para  $i, j = 1, \dots, n$ . Desarrollar un algoritmo que encuentre un emparejamiento óptimo entre hombres y mujeres, en el sentido de que la suma de los productos de las preferencias sea máximo.

**Ejercicio 4** Aplicar ramificación y poda al problema del coloreado óptimo de un grafo no dirigido. Se han de pintar los nodos de un grafo con una serie de colores, de manera que ningún arco del mismo una dos nodos del mismo color. Se trata de hacerlo utilizando el mínimo número de colores posible.

**Ejercicio 5** Tenemos que distribuir a  $n$  invitados en una mesa redonda con espacio para  $n$  comensales. Se dispone de una función  $afinidad(i, j)$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $i \neq j$ , que devuelve un valor positivo según el grado de afinidad que los invitados  $i$  y  $j$  tienen entre sí (a mayor valor, mayor afinidad). Diseñar un algoritmo que calcule la distribución de los invitados en la mesa de forma que se maximice el bienestar general. El bienestar general se calcula sumando las afinidades de los comensales sentados en posiciones adyacentes.

**Ejercicio 6** Se desea organizar el banquete de una boda para  $n$  comensales. Se dispone de un comedor con  $m$  mesas de capacidad  $n/m$ , siendo  $n$  múltiplo de  $m$ . Se conocen las afinidades  $a_{ij}$ ,  $1 \leq i, j \leq n$  entre cada par de comensales. La matriz de afinidades no es necesariamente simétrica, aunque sabemos que  $a_{ii} = 0$ , para todo  $i$ . Definimos la satisfacción global  $S$  de una distribución de comensales en mesas como la suma  $S = \sum_{k=1}^m \sum_{i,j \in mesa_k} a_{ij}$ . Diseñar un algoritmo que encuentre una asignación de comensales a mesas que maximice la satisfacción global. Se prestará especial atención a no generar soluciones equivalentes.

**Ejercicio 7** Tenemos  $n$  trabajos que ejecutar y  $m$  procesadores iguales que trabajan en paralelo (con  $m < n$ ). El tiempo necesario para ejecutar el trabajo  $i$  en cualquiera de los procesadores es  $t_i$ . Teniendo en cuenta que el orden en el que se ejecuten los trabajos dentro de un mismo procesador no es significativo, escribe un algoritmo utilizando la técnica de ramificación y poda que determine qué trabajos se deben ejecutar en qué procesadores de tal forma que se minimice el tiempo de finalización del último trabajo que se ejecuta. Hay que evitar encontrar soluciones equivalentes: dos soluciones se consideran equivalentes si una se consigue a partir de la otra simplemente mediante la permutación de los procesadores.

**Ejercicio 8** Deseamos decorar una pared de  $L$  metros de ancho. Hemos tenido la innovadora idea de hacerlo colgando una hilera de cuadros pegados lado con lado. Nos disponemos a comprar los cuadros en la feria de arte moderno, donde tenemos la posibilidad de elegir entre  $n$  cuadros. Cada cuadro tiene un prestigio  $p_i$ , y unas dimensiones de  $a_i$  metros de alto por  $b_i$  metros de ancho,  $1 \leq i \leq n$ . Dado lo peculiar de los cuadros, podemos elegir colgar cada cuadro tanto en horizontal como en vertical sin que por ello se vea afectado su prestigio. Lo que no podemos hacer es trocear un cuadro. Diseñar un algoritmo que determine qué cuadros comprar de forma que la longitud de la hilera de cuadros sume exactamente  $L$  metros y se maximice el prestigio acumulado en la pared.

**Ejercicio 9** Dado un grafo valorado no dirigido  $G = \langle V, A \rangle$  y un entero  $r$ , escribir un algoritmo que encuentre, si existe, un cliqué de tamaño  $r$  y de mínimo coste en  $G$ . Un cliqué es un subgrafo completo, es decir, un subconjunto  $V' \subseteq V$  de vértices tal que existe una arista entre cada par de vértices en  $V'$ . El tamaño de un cliqué es su número de vértices, y su coste es la suma de los costes de sus aristas. Si hubiese empates basta dar uno de ellos.

**Ejercicio 10** Dado un grafo no dirigido y no valorado  $G = \langle V, A \rangle$ , un subconjunto de vértices  $V' \subseteq V$  es independiente si no hay dos vértices en  $V'$  unidos por una arista en  $G$ . Escribir un algoritmo que devuelva el tamaño del mayor conjunto independiente de vértices de un grafo dado  $G$ .

**Ejercicio 11** Tenemos un conjunto de  $n$  componentes electrónicas para colocar en  $n$  posiciones sobre una placa. Se nos dan dos matrices de dimensiones  $n \times n$ ,  $N$  y  $D$ , donde  $N[i, j]$  indica el número de conexiones necesarias entre la componente  $i$  y la componente  $j$ , y  $D[p, q]$  indica la distancia sobre la placa entre la posición  $p$  y la posición  $q$  (ambas matrices son simétricas y con diagonales nulas). Un cableado  $(x_1, \dots, x_n)$  de la placa consiste en la colocación de cada componente  $i$  en una posición distinta  $x_i$ . La longitud total de este cableado viene dada por la fórmula  $\sum_{i < j} N[i, j]D[x_i, x_j]$ . Escribir un algoritmo para encontrar el cableado de longitud mínima.

**Ejercicio 12** Una de las pruebas habituales del concurso *Supervivientes* es la construcción de una cabaña rudimentaria cuyo techo es una simple lona soportada por cuatro pilares. Los concursantes dispondrán de  $n$  de fragmentos de caña de longitudes enteras  $l_1, \dots, l_n$  ensamblando los cuales deberán obtener los cuatro pilares. El objetivo es que sus alturas queden razonablemente equilibradas y sean lo más altas posibles. Precisando, desarrollar el problema buscando maximizar el pilar más bajo de los cuatro.

**Ejercicio 13** El Decanato de la Facultad de Informática quiere celebrar una macrofiesta para todo el personal docente y administrativo y los alumnos. Aprovechando el buen tiempo, la fiesta tendrá lugar en el jardín. Se prevé que la fiesta se prolongue hasta horas nocturnas. Para evitar sorpresas (desagradables) se quieren instalar focos suficientes para iluminar todo el jardín. A tal efecto se han recabado todas las existencias de bombillas de 200 w disponibles en el almacén de la UCM, pero como dichas bombillas pertenecen a marcas, calidades y remesas diferentes, cada una tiene una duración (en tiempo) distinta. El objetivo es mantener una iluminación de un total de 2 Kw durante el mayor tiempo posible.

**Ejercicio 14** Aplicar ramificación y poda al problema de la confección de horarios con restricciones: Se han de impartir una serie de clases y para cada una de ellas se cuenta con una serie de horarios en las cuales pueden ser impartidas. Se trata de impartir todas ellas utilizando el menor número posible de aulas.

**Ejercicio 15** Alonso Rodríguez tiene que hacer la compra de la semana consistente en  $n$  productos. En su barrio hay  $m$  supermercados, cada uno de los cuales dispone de todos esos productos. Pero no quiere comprar más de tres productos en cada uno de los supermercados, ya que así pasa más tiempo comprando (se puede suponer que  $n \leq 3m$ ). Dispone de una lista de precios de los productos en cada uno de los supermercados. Escribir un algoritmo de ramificación y poda, aplicando el esquema optimista-pesimista en caso de ser posible, que le permita decidir cómo hacer la compra de forma el coste total sea mínimo.

**Ejercicio 16** Juanita afronta sus exámenes finales del curso con precipitación. Para cada asignatura  $A_i$  del curso conoce los días  $f_i$  que quedan hasta la fecha de su examen, y sabe la cantidad de días  $d_i$  que debería dedicar a su estudio (lógicamente, antes del correspondiente examen) para poder aprobarla. Juanita se teme que no podrá con todas las asignaturas, así que ha dado un valor  $v_i$  a cada una de ellas, y ha de determinar qué asignaturas estudiará, y en qué orden, hasta la finalización de los exámenes, a sabiendas de que si dedica a una asignatura menos de los  $d_i$  estipulados la suspenderá irremediamente. Debéis ayudar a Juanita a elegir las asignaturas que estudiará, de manera que maximice el valor total de las asignaturas que ha decidido estudiar con tiempo suficiente para aprobarlas. Describir el algoritmo de ramificación y poda apropiado para este problema, e indicar con detalle las cotas inferior y superior empleadas.

**Ejercicio 17** Para un conjunto de  $n$  hombres y otro de  $n$  mujeres, nos dan dos matrices de números naturales  $n \times n$ ,  $H$  y  $M$ , en las que  $H_{ij}$  representa la afinidad del hombre  $i$  hacia la mujer  $j$  y  $M_{ij}$

la afinidad de la mujer  $i$  hacia el hombre  $j$ . Un emparejamiento es cualquier aplicación biyectiva entre ambos conjuntos. Un emparejamiento es estable si en el mismo no existe un hombre  $i$  y una mujer  $j$  tales que  $i$  tenga más afinidad por  $j$  que por su pareja y a la vez  $j$  tenga más afinidad por  $i$  que por la suya. El valor de un emparejamiento estable es la suma de los productos de las afinidades mutuas de cada pareja. Se pide desarrollar un algoritmo que devuelva el emparejamiento estable de mayor valor.

**Ejercicio 18** La Facultad de Informática debe planificar sus exámenes de septiembre de 2020 necesariamente entre los días 1 y  $n$  de ese mes. Tiene que realizar  $m$  exámenes,  $m > n$ , y puede planificar cada día cuantos exámenes desee, siempre que no sean incompatibles entre sí, y para ello dispone de una matriz  $I$ ,  $m \times m$ , de booleanos, donde  $I_{ij} = \text{true}$  si los exámenes  $i$ ,  $j$  son incompatibles e  $I_{ij} = \text{false}$  en caso contrario.

Implementar un algoritmo que indique el día a realizar cada examen, de forma que el máximo número de aulas necesario sea mínimo. Se supondrá que cada examen ocupa un aula durante todo el día. Se valorará negativamente el que se generen soluciones equivalentes.

**Ejercicio 19** Se quiere grabar en vídeo un concierto al aire libre desde tres puntos distintos, para lo que se cuenta con tres cámaras de la misma marca y modelo. Debido a que no hay puntos de electricidad a los que enchufar las mismas, los organizadores se han visto obligados a utilizar  $n$  baterías. Como las han comprado en diversas tiendas, sus duraciones  $d_i$ ,  $1 \leq i \leq n$ , varían de unas a otras. El problema consiste en decidir cómo distribuir las baterías entre las tres cámaras de manera que se maximice el tiempo que las tres estén simultáneamente grabando.

**Ejercicio 20** Alonso Rodríguez tiene que hacer la compra de la semana consistente en  $n$  productos. En su barrio hay  $m$  supermercados, cada uno de los cuales dispone de todos ellos, pero no quiere comprar más de tres productos en cada establecimiento. Se puede suponer que  $n \leq 3m$ . Dispone de una tabla  $P$ ,  $n \times m$ , de precios de cada producto en cada uno de los supermercados. Diseñar un algoritmo de ramificación y poda que le permita decidir cómo hacer la compra de forma que el coste total sea mínimo.