Métodos Algorítmicos en Resolución de Problemas I

Grado en Ingeniería Informática
Doble Grado en Ingeniería Informática y Matemáticas

Hoja de ejercicios 5 | Curso 2024–2025

Ejercicios del método voraz

Ejercicio 1 El siguiente problema aparece en el análisis automático de programas: dado un conjunto de n variables $x_1, \ldots, x_n$, se tienen algunas restricciones de igualdad de la forma $x_i = x_j$ y algunas de desigualdad de la forma $x_i \neq x_j$. En general, no será posible satisfacer todas; por ejemplo,

$$x_1 = x_2,\ x_2 = x_3,\ x_3 = x_4,\ x_1 \neq x_4$$

no se pueden satisfacer. Escribir un algoritmo eficiente que reciba m restricciones sobre n variables y decida si son o no satisfactibles.

Ejercicio 2 En una cinta magnética hay que grabar n programas de longitudes $l_1, \ldots, l_n$. Se supone que la velocidad de lectura es constante, y que tras cada búsqueda seguida de la lectura de un programa, la cinta es automáticamente rebobinada. Se conoce la tasa de utilización de cada programa, esto es, se sabe que, del número total de peticiones, un porcentaje p_i corresponde al programa $i, 1 \leq i \leq n$, siendo $\sum_{i=1}^{n} p_i = 1$. El objetivo es minimizar el tiempo medio de carga, el cual es proporcional a

$$T = \frac{1}{n} \sum_{j=1}^{n} \left(p_{i_j} \sum_{k=1}^{j} l_{i_k} \right)$$

cuando los programas están almacenados en el orden $i_1, i_2, \ldots, i_n$.

1. Demostrar mediante un contraejemplo que la secuencia en orden creciente de l_i no es necesariamente óptima.
2. Demostrar asimismo que la secuencia en orden decreciente de p_i tampoco lo es.
3. Demostrar finalmente que la secuencia ordenada en orden decreciente de p_i/l_i es óptima.

Ejercicio 3 Un viajante de comercio tiene que viajar en coche desde Valencia a Lisboa siguiendo una ruta preestablecida. Con el déposito lleno, su coche puede recorrer un máximo de n kilómetros. El viajante dispone de un mapa de carreteras en el cual figuran las distancias entre gasolineras en su ruta y desea utilizar esa información para realizar un número mínimo de paradas para repostar combustible en su recorrido. Para ello hay que desarrollar un método eficiente para determinar en qué gasolineras tiene que parar, implementar el método y demostrar que esa estrategia da lugar a una solución óptima.

Ejercicio 4 Dados n números reales, con n par, emparejarlos de tal forma que al obtener la suma de los números de cada pareja, se minimice la suma máxima.

Ejercicio 5 Se tienen dos vectores, uno B de bases y otro E de exponentes, ambos consistentes en n enteros positivos. Se pide emparejar cada base de B con un exponente distinto de E de forma que, si $\{(b_i, e_i) \mid 1 \leq i \leq n\}$ son las parejas formadas, el producto $\prod_{i=1}^{n} b_i^{e_i}$ sea máximo. Demostrar la optimalidad del emparejamiento propuesto.

Ejercicio 6 En la tercera planta del Hospital Bajomínimos conocen la hora y minuto exactos en que han de pinchar a sus enfermos. La duración del turno de cada enfermera es de k horas.

1. Escribe un algoritmo eficiente que calcule el intervalo de k horas durante el que será necesario poner más inyecciones. Iterando el mismo, puedes obtener una planificación que te indique a qué hora debería comenzar el turno de cada enfermera a contratar, de manera que finalmente queden cubiertas todas las inyecciones.

2. Razonar que ese algoritmo no genera siempre la mejor planificación posible (la que menos enfermeras necesita).
3. Escribe otro algoritmo que sí lo haga y demuestra rigurosamente su corrección. ¿Cuál es su coste en tiempo? Compáralo con el de la solución alternativa arriba descartada.

Ejercicio 7 Disponemos de una cinta de tela dividida a lo largo en N cuadraditos. Sobre cada uno de ellos, i, con $i \in \{1, \dots, N\}$, aparece impreso un número natural $c_i \geq 0$. Tenemos a nuestra disposición una cantidad ilimitada de tiritas iguales, de la anchura de la cinta y longitud constante k, con $1 \leq k \leq N$, que podemos pegar sobre la cinta cubriendo k cuadraditos consecutivos de la misma. Al pegar las tiritas podemos solaparlas arbitrariamente unas sobre otras. Decimos que un cubrimiento de la cinta con tiritas es *satisfactorio* cuando cada cuadradito i quede cubierto por al menos c_i tiritas.

Se debe encontrar un cubrimiento satisfactorio óptimo, que utilice el mínimo número posible de tiritas.

1. Diseñar un algoritmo devorador que genere un cubrimiento óptimo.
2. Programarlo con suficiente detalle e indicar su complejidad.
3. Demostrar con suficiente detalle que el algoritmo propuesto es correcto.

Ejercicio 8 La Universidad Abierta de Pernambuco debe examinar presencialmente a sus alumnos. Para ello dispone de una cantidad limitada m de aulas, para las que conocemos sus capacidades $C_1, \dots, C_m$. La universidad cuenta con alumnos (diferentes unos de otros) matriculados en n asignaturas distintas, habiendo A_j alumnos en cada una de ellas, con $j \in \{1, \dots, n\}$. Aunque se sabe que en un mismo día será difícil examinarlos a todos, se desea convocar los exámenes de un conjunto de asignaturas que sí puedan celebrarse conjuntamente, con la suposición de que en cada aula solo podremos examinar a alumnos de una misma asignatura. El objetivo es examinar ese día al máximo número posible de estudiantes.

Diseñar un algoritmo voraz y razonar su corrección, con la restricción adicional de que todos los alumnos de una asignatura han de ser examinados en la misma aula.

Ejercicio 9 En época de rebajas, es frecuente encontrar en muchos comercios ofertas 3×2 en la que por cada tres objetos comprados te regalan uno. Los regalados son los de menor precio. Por ejemplo, si se compran zapatillas con precios 40, 35, 30, 25, 20 y 15 euros, te descuentan un total de 35 euros. Pero a veces conviene hacer más de una compra para obtener un descuento mayor. En el ejemplo anterior, haciendo una primera compra con las zapatillas de 40, 30, 25 y una segunda compra con las de 35, 20 y 15 euros, se obtiene un ahorro total de 40 euros.

Dados una cantidad de objetos múltiplo de 3 y los precios de los objetos, diseñar un algoritmo voraz para organizar las compras de forma que se maximice el ahorro total, y razona formalmente que el algoritmo calcula realmente el óptimo.

Ejercicio 10 El espacio televisivo *Masterchef* ha decidido organizar un banquete por todo lo alto para conmemorar su programa número 25.000, tras mantenerse 500 años en los máximos niveles de audiencia. Para tal evento dispone de un conjunto de m mesas con capacidades diversas $M_1, \dots, M_m$.

Un conjunto de n grupos de tamaños $G_1, \dots, G_n$ han solicitado acudir al banquete. Para seleccionar entre ellos a los agraciados, los criterios del director del programa son tres:

1. No se puede partir ningún grupo entre varias mesas.
2. No se puede compartir una mesa entre varios grupos.
3. Desea maximizar el número total de comensales del banquete.

Diseñar un algoritmo voraz para realizar tal selección y razona formalmente que calcula realmente el óptimo.

Ejercicio 11 Textiles S. A. ha recibido el encargo de fabricar n alfombras rojas de longitudes $l_1, \ldots, l_n$ para otras tantas bodas. Con las prisas, no se tomaron bien las medidas de los pasillos de las respectivas iglesias y algunas alfombras han quedado cortas. Las medidas reales de los pasillos son $p_1, \ldots, p_n$.

1. Diseña un algoritmo voraz que asigne una alfombra a cada iglesia, de forma que la suma de los metros de los pasillos que quedan totalmente cubiertos por su alfombra sea máxima y demuestra su corrección.
2. Una pareja de novios queda satisfecha con la alfombra si su longitud cubre completamente el pasillo e incluso si sobra algo de alfombra. Diseña una estrategia voraz que asigne una alfombra a cada iglesia de forma que se maximice el número de parejas de novios satisfechas y demuestra su corrección.

Ejercicio 12 Nos dan un conjunto de n tareas, cada una con una hora de comienzo c_i y una hora de terminación f_i, $1 \leq i \leq n$, que han de ser realizadas por un mismo procesador. Dos tareas son compatibles ni no se solapan en el tiempo. Se desea un algoritmo voraz que escoja un subconjunto de tareas, compatibles entre sí, de cardinal máximo. Se pide:

1. Mostrar mediante un contraejemplo que la estrategia de elegir en cada iteración la tarea que comienza más temprano y que no solapa con las anteriores no es óptima.
2. Mostrar mediante un contraejemplo que la estrategia de elegir en cada iteración la tarea más corta que no solapa con las anteriores no es óptima.
3. Mostrar mediante un contraejemplo que la estrategia de elegir en cada iteración la tarea con menor número de conflictos no es óptima.
4. Diseñar e implementar una estrategia óptima, y razonar que realmente lo es.

Ejercicio 13 La fábrica *Low-Cost Umbrellas* necesita varillas de longitud L para sus paraguas y dispone de un conjunto de n varillas de longitudes $l_1, \ldots, l_n$, todas ellas más cortas que L. Se sabe que se pueden soldar dos varillas cortas para formar una más larga sin que se resienta la calidad de los paraguas. Diseñar una estrategia voraz y demostrar su corrección para emparejar las varillas que se consideren relevantes de forma que se maximice el número de varillas soldadas que alcanzan una longitud mayor o igual que L.

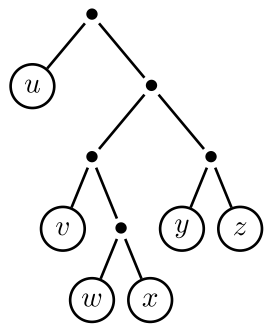
Ejercicio 14 En una estación de tren se detienen al día n trenes. Se conocen las horas de llegada a la estación y de salida de la misma de dichos trenes. Implementar un algoritmo voraz que calcule el número mínimo de vías necesarias en la estación para atender a los n trenes. Demostrar que la estrategia elegida es óptima. ¿Cuál es el coste del algoritmo? Se valorarán mejor los algoritmos que tengan un coste en $O(n \log n)$.

Ejercicio 15 Supongamos que un fichero contiene caracteres de 8 bits tales que sus frecuencias son aproximadamente las mismas: en concreto, la mayor frecuencia es inferior al doble de la frecuencia mínima. Demostrar que la codificación de Huffman en este caso no resulta más eficiente que una codificación ordinaria de longitud fija igual a 8.

Ejercicio 16 Demostrar que si en un árbol binario algún nodo no tiene dos hijos entonces no puede corresponder a una codificación sin prefijos óptima.

Ejercicio 17 Dar una codificación de Huffman para una tabla de frecuencias que coincide con los primeros 8 números de la sucesión de Fibonacci. Generalizar la respuesta, razonando la solución, cuando la tabla de frecuencias coincide con los n primeros números de Fibonacci.

Ejercicio 18 Dado un fichero en el que aparecen los caracteres u, v, w, x, y y z con frecuencias f_u, f_v, f_w, f_x, f_y y f_z, respectivamente, el algoritmo de Huffman ha calculado el siguiente árbol:



Conjeturar y justificar valores posibles para dichas frecuencias como porcentajes de la longitud del fichero. Indicar también el porcentaje de compresión del fichero que se obtendrá frente a una codificación de longitud fija de 3 bits.

Ejercicio 19 Se desea organizar una competición en la que participan k equipos, cada uno de los cuales cuenta con S_i seguidores, $1 \leq i \leq k$. Cada vez que se juega un partido, el equipo que gana sigue compitiendo, y el perdedor desaparece de la competición. Cada vez que un equipo pierde, todos sus seguidores pasan a serlo del equipo que los ha derrotado. A cada partido acuden todos los seguidores (actuales) de los dos equipos. Se pide:

1. Diseñar un algoritmo que organice los partidos (es decir, que decida quién debe jugar con quién) de forma que la suma de las asistencias a los mismos sea mínima. Demostrar que la solución computada es en efecto la que produce la mínima asistencia global.
2. Diseñar otro algoritmo que organice la competición de forma que ahora la suma de las asistencias a los partidos celebrados sea máxima. Demostrar que la solución computada es en efecto la que produce la máxima asistencia global.

Ejercicio 20 Un viajero debe realizar una ruta por el desierto de Kalahari sin morir de sed. Conoce la posición de $n \geq 2$ oasis en los que puede parar a rellenar su cantimplora de C litros de capacidad con agua fresca, y sabe que su consumo de agua es de L litros por kilómetro. Los oasis están numerados de 1 a n, siendo el oasis 1 el de salida, en el que llena completamente su cantimplora, y n el de destino. El vector $D[1..n]$ indica para cada oasis i, la distancia en kilómetros desde el oasis 1 hasta el oasis i. Se pide idear e implementar una estrategia voraz para decidir las paradas que debe hacer de forma que el número de paradas sea mínimo, y demostrar la optimalidad de dicha estrategia.

Ejercicio 21 En la playa de Malibú tenemos N puestos de vigilancia y N socorristas. Los puestos de vigilancia se encuentran situados a lo largo de la orilla a distancias medidas desde uno de sus extremos, que están guardadas en un vector D de distancias. Durante la mañana los socorristas han estado paseando por la orilla para vigilar a los bañistas pero al final de la mañana han de acudir cada uno a uno de los puestos de vigilancia para hacer un informe (en cada puesto solo cabe un socorrista). Las posiciones de los socorristas en la orilla están medidas desde el mismo extremo de la playa que los puestos y guardadas en un vector S. Cada socorrista tarda tantos minutos como distancia hay entre él y el puesto al que acude. Por ejemplo, si un puesto está en la posición 0 y un socorrista en la 2, tardará 2 minutos en llegar a ese puesto. El coordinador ha de esperar a que el último termine su informe para poder irse a casa y está deseando llegar para ver a su familia. Para ello, idea una estrategia para indicar a cada socorrista a qué puesto acudir a hacer su informe de forma que el coordinador pueda irse a casa lo mas pronto posible, y demuestra que dicha estrategia es óptima.