Métodos Algorítmicos en Resolución de Problemas I

Grado en Ingeniería Informática

Hoja de ejercicios 5 | Curso 2020-2021

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 En el problema de la mochila, demostrar mediante un contraejemplo que si se añade la condición de que los objetos no se pueden partir, entonces la estrategia devoradora de escoger en orden decreciente de valor por unidad de peso no es necesariamente óptima.

Supongamos que, en vez de n objetos, tenemos n tipos de objetos (que se pueden partir). Formalmente, la restricción $0 \leq x_i \leq 1$ se convierte en $0 \leq x_i$. ¿Funciona correctamente en este contexto el algoritmo devorador basado en el precio por unidad?

Ejercicio 4 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 5 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 6 La Universidad Imponente tiene que planificar un evento cultural que consiste en n conferencias. Para cada conferencia se conoce la hora de comienzo y la de finalización, fijadas por los ponentes. Se ha pedido al Departamento de Informática que planifique las n conferencias distribuyéndolas entre las distintas salas disponibles, de forma, claro está, que no haya dos conferencias en una misma sala al mismo tiempo. El objetivo es minimizar el número de salas utilizadas, para así causar el menor trastorno al resto de las actividades académicas.

Ejercicio 7 La filmoteca ha organizado un maratón de cine de terror. Durante 24 horas se proyectarán películas (todas diferentes) en las n salas disponibles. Deborah Cinema, gran aficionada a este género de películas, ha conseguido la programación completa donde aparecen todas las películas que se van a proyectar durante el maratón. Junto con el título, nombre del director, duración de la película y otros datos de interés, se indica la sala de proyección y la hora de comienzo. Ayudar a Deborah a planificar su maratón de cine, teniendo en cuenta que su único objetivo es ver el máximo número posible de películas.

Ejercicio 8 Se dispone de un conjunto finito $M = \{m_0, m_1, \ldots, m_n\}$ de tipos de monedas, indicando cada m_i el valor de las monedas de tipo i. Dichos valores son todos distintos y están ordenados crecientemente; supondremos $m_0 = 1$. Suponiendo que se dispone de una cantidad limitada l_i de monedas de cada tipo i, diseñar un algoritmo que, dada un cantidad $C > 0$, reúna dicha cantidad con el menor numero posible de monedas.

1. Indicar si el problema tiene siempre solución.
2. Si cada tipo es un múltiplo del anterior, demostrar si la estrategia voraz consistente en escoger las monedas por tipos decrecientes es o no óptima.

Ejercicio 9 A los hinchas de los Broncos de Boston les encanta aplastar a sus rivales hasta el punto de medir el éxito de sus ligas por la suma total de las diferencias de puntos de las victorias que logren en su estadio. Es decir, si en un partido i, b_i son los puntos de los Broncos y r_i son los puntos de su rival, ellos pretenden maximizar la suma de las $b_i - r_i$ que cumplan $b_i > r_i$.

Consultando a un adivino, los Broncos saben la secuencia exacta $[r_1, \ldots, r_n]$ de los puntos que conseguirán sus rivales en los próximos n partidos. También saben el conjunto exacto $\{b_1, \ldots, b_n\}$ de sus propios puntos en esos partidos, pero han de decidir en qué orden los conseguirán.

1. Proporciona un algoritmo voraz que decida la permutación óptima de dichas anotaciones.
2. Demuestra, con el máximo rigor posible, que tu solución es en efecto la óptima.

Ejercicio 10 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 11 Demostrar que si en un árbol binario algún nodo no tiene dos hijos entonces no puede corresponder a una codificación sin prefijos óptima.

Ejercicio 12 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 13 Un *conjunto de aristas recubridoras* de un grafo no dirigido $G = (V, A)$ es un conjunto de aristas $A' \subseteq A$ que interseca cada ciclo del grafo. Así, si se eliminan las aristas A' se obtiene un grafo acíclico. Escribir un algoritmo eficiente que dado un grafo valorado, no dirigido y conexo, devuelva el conjunto de aristas recubridoras de menor coste.

Ejercicio 14 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 15 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 16 Disponemos de una cinta de tela dividida a lo largo en N cuadraditos. Sobre cada uno de ellos, i, con $i \in \{1, \ldots, N\}$, 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 17 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 $C_1, \ldots, C_m$. La universidad cuenta con alumnos (diferentes unos de otros) matriculados en n asignaturas distintas, habiendo A_j alumnos en cada una de ellas, con $j \in \{1, \ldots, n\}$. 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 18 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 19 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 $M_1, \ldots, M_m$.

Un conjunto de n grupos de tamaños $G_1, \ldots, G_n$ 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 nmero total de comensales del banquete.

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

Ejercicio 20 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 $p_1, \ldots, p_n$. 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 las alfombras a las iglesias de forma que se maximice el número de parejas de novios satisfechas y demuestra su corrección.

Ejercicio 21 La Conferencia Episcopal ha encargado a Textiles S. A. fabricar n alfombras de longitudes $l_1, \ldots, l_n$ para sus iglesias. 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 $p_1, \ldots, p_n$. 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. Demuestra su corrección.

Ejercicio 22 Sean $f_1, \ldots, f_n$ las fuerzas en Kg de n bueyes. Tenemos que formar parejas disjuntas de bueyes para tirar de un carro de peso P. Una pareja (i, j) es válida si $f_i + f_j > P$. Diseñar una estrategia voraz que maximice el número de parejas válidas. No es necesario programar el algoritmo, pero sí indicar la forma de la solución devuelta y el razonamiento de corrección.

Ejercicio 23 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. Diseñar e implementar una estrategia óptima, y razonar que realmente lo es.

Ejercicio 24 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 25 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 $O(n \log n)$.