UNIVERSIDAD DE MÁLAGA

Análisis y Diseño de Algoritmos
Algoritmos Voraces
(2º de Grado Ing. Inf., Ing. Sw., Ing. Comp.)
E.T.S.I. INFORMÁTICA

Ejercicios Básicos

Para cada uno de los siguientes problemas:

- Compruebe si se cumple el principio de subestructura óptima.
- Diseñe un enfoque voraz para resolverlo, e impleméntelo en pseudocódigo.
- Demuestre que el algoritmo voraz diseñado encuentra la solución óptima, o muestre un contra-ejemplo en el que no la encuentra.

1. Dada una fracción a/b se desea expresarla como la suma de un número mínimo de fracciones distintas con numerador unidad, e.g., $3/5 = 1/2 + 1/10$.
2. Durante la preparación de la semana cultural de Informática se recibe un conjunto de solicitudes para usar el salón de actos. Cada una de estas solicitudes requiere uso exclusivo del salón, y está determinada por una hora de comienzo c_i y una hora de finalización f_i. Determinar el mayor número de solicitudes que se pueden satisfacer.
3. Dado un grafo $G(V, E)$ se desea encontrar un conjunto $S \subseteq V$ de tamaño mínimo tal que cada arista $(u, v) \in E$ tiene al menos uno de sus extremos en S.
4. Dado un grafo $G(V, E)$, un *coloreado* del mismo consiste en asignar un color a cada vértice de manera que si $(u, v) \in E$ entonces u y v tienen distinto color. Se desea encontrar un coloreado que emplee el mínimo número de colores.
5. Vamos a ir en coche desde Málaga a Bilbao a ver el Guggenheim. Salimos con el depósito lleno, y sabemos que con ese combustible podemos recorrer M kilómetros. Tenemos un mapa en el que figuran los puntos kilométricos $k(1), \cdots, k(n)$ en los que hay gasolineras, y queremos determinar en cuáles de ellas hay que repostar para llegar al destino haciendo el menor número de paradas.
6. Tenemos que planificar la gira veraniega del grupo de teatro de Informática. Tenemos n propuestas, cada una de las cuales consta de una fecha de inicio d_i, un número de actuaciones a_i (siempre una al día), y una oferta económica m_i. Seleccionar las propuestas que aceptaremos para maximizar el beneficio económico.
7. Definamos un *intervalo unitario* como un subconjunto de la recta real de la forma $[a, b] = \{z \mid a \leq z \leq b\}$, donde $b - a = 1$. Por ejemplo, $[\frac{5}{4}, \frac{9}{4}]$ es un intervalo unitario. Considere el *problema del cubrimiento de intervalo* donde la entrada es un conjunto $\{x_1, x_2, ..., x_n\}$ de n puntos en un orden arbitrario en la recta real, y la salida es un conjunto formado por el menor número de intervalos unitarios que cubren todos los puntos de entrada (en el sentido de que cada punto está en al menos uno de los intervalos). Proponga un algoritmo voraz para resolver el problema (de forma correcta si es posible)

   Como un ejemplo, supongamos que la entrada es $\{\frac{7}{4}, \frac{7}{2}, \frac{1}{2}, 2, \frac{3}{2}, 0\}$. Entonces $\{[0, 1], [\frac{5}{4}, \frac{9}{4}], [\frac{5}{2}, \frac{7}{2}]\}$ es una salida (i.e., respuesta) válida al problema planteado pues se puede comprobar fácilmente que estos tres intervalos cubren todos los seis puntos de entrada, y también que no existen dos intervalos unitarios que cubran todos los puntos de entrada.
8. Se tienen n procesos, cada uno de los cuales tiene duración t_i, que se desean ejecutar en un procesador. Dado un cierto orden de ejecución de los procesos, el tiempo de estancia en el sistema de cada uno de los mismos es el tiempo de espera hasta ser ejecutados más el tiempo de ejecución. Se desea encontrar un orden de ejecución que minimice la suma de los tiempos de estancia en el sistema de todos los procesos.

9. El *problema de la mochila 0-1 multidimensional* generaliza al problema de la mochila 0-1 como sigue: se tienen n objetos con valores v_i; se tienen m mochilas con capacidades c_j; el objeto i-ésimo ocupa un volumen a_{ij} en la mochila j-ésima. Se desea encontrar el subconjunto S de objetos que cabe simultáneamente en las m mochilas, i.e.,

$$\forall j \ (1 \leqslant j \leqslant m) : \sum_{i \in S} a_{ij} \leqslant c_j$$

tal que se maximiza $\sum_{i \in S} v_i$.

10. Dado un conjunto de números naturales $S = \{s_1, \cdots, s_n\}$ se desea encontrar una partición en dos subconjuntos disjuntos S_1 y S_2 (i.e., $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \emptyset$) tal que se minimiza

$$\left| \sum_{x \in S_1} x - \sum_{y \in S_2} y \right|$$

Ejercicios Complementarios

1. Consideremos n programas $P_1, P_2, \ldots P_n$ que debemos almacenar en un disco. El programa P_i requiere s_i megabytes de espacio en disco, y la capacidad del disco es de D megabytes.

   a) Se desea maximizar el número de programas almacenados en el disco. Demostrar lo siguiente o dar un contraejemplo: podemos utilizar un algoritmo ávido que seleccione los programas por orden creciente de s_i.

   b) Se desea maximizar el espacio utilizado del disco. Demostrar lo siguiente o dar un contraejemplo: se puede utilizar un algoritmo ávido que seleccione los programas por orden decreciente de s_i.

2. Supongamos que disponemos de n trabajadores y n tareas. Sea $b_{ij} > 0$ el coste de asignarle el trabajo j al trabajador i. Una asignación de tareas puede ser expresada como una asignación de los valores 0 ó 1 a las variables x_{ij}, donde $x_{ij} = 0$ significa que al trabajador i no le han asignado la tarea j, y $x_{ij} = 1$ indica que sí. Una asignación válida es aquella en la que a cada trabajador sólo le corresponde una tarea y cada tarea está asignada a un trabajador. Dada una asignación válida, definimos el coste $C(x)$ de dicha asignación como:

$$C(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} b_{ij}$$

   Diremos que una asignación es óptima si es de mínimo coste. Demostrar o dar un contraejemplo que determine si alguna de las estrategias propuestas encuentra siempre solución óptima

   a) asignar cada trabajador la mejor tarea posible

   b) asignar cada tarea al mejor trabajador disponible

3. Dado el problema del cambio de dinero, probar o dar un contraejemplo para las siguientes afirmaciones:

   a) Suponiendo que cada moneda del sistema monetario del país vale al menos el doble que la moneda de inferior valor, que existe una moneda de valor unitario, y que disponemos un número ilimitado de monedas de cada valor, entonces el algoritmo ávido más simple encuentra siempre una solución óptima.

   b) Suponiendo que el sistema monetario estuviera compuesto por monedas de valor 1, p, p^2, p^3, $\ldots$, p_n, donde $p > 1$ y $n > 0$, y que dispusiéramos de un número ilimitado de monedas de cada valor, entonces el algoritmo ávido más simple encuentra siempre una solución óptima.

4. Supongamos que el coste de tender una línea de teléfono entre dos puntos a y b es directamente proporcional a su distancia euclídea, y que necesitamos conectar telefónicamente varias ciudades con un coste mínimo. Encontrar un ejemplo en donde cueste menos establecer una centralita en un punto situado entre todas las ciudades que conectarlas todas usando enlaces directos entre ellas.

5. ¿Qué ocurriría al ejecutar los algoritmos de Prim y Kruskal que hemos implementado si por error les suministráramos un grafo no conexo? Analizar también la complejidad (espacio y tiempo) de ambos algoritmos. ¿Cómo podrían mejorarse?

6. Para resolver el algoritmo de recorrer el tablero de ajedrez con un caballo, un posible algoritmo ávido decide, en cada iteración, colocar el caballo en la casilla desde la cual domina el menor número posible de casillas aún no visitadas. Impleméntese este algoritmo y demuéstrese su validez.

7. Dado un grafo dirigido, se llama camino hamiltoniano a aquel que visita exactamente una vez cada nodo del grafo y no vuelve al punto de partida. Demostrar que todo grafo dirigido fuertemente conexo posee siempre un camino hamiltoniano. Proponer un algoritmo ávido que lo encuentre.

8. Se conocen las distancias entre un cierto número de ciudades. Un viajante debe, a partir de una de ellas, visitar cada ciudad exactamente una vez y regresar al punto de partida habiendo recorrido en total la menor distancia posible. Un posible algoritmo ávido para solucionar este problema escogería en cada iteración el arco más corto aún no considerado que cumpliera las dos condiciones siguientes: (a) no formar un ciclo con los arcos ya seleccionados, excepto en la última iteración, que es donde completa el viaje; y (b) no es el tercer arco que incide en de mismo nodo de entre los ya escogidos. Se pide implementar el algoritmo.

9. Dada una secuencia de palabras $p_1, p_2, \ldots, p_k$ de longitudes $l_1, l_2, \ldots, l_k$ se desea agrupar en líneas de longitud L. Las palabras están separadas por espacios cuya amplitud es b, pero los espacios pueden reducirse o ampliarse si es necesario (aunque sin solapamiento de palabras), de tal forma que una linea $p_i, p_{i+1}, \ldots, p_j$ tenga exactamente longitud L. Sin embargo, la multa por reducción o ampliación es el número total de espacios que aparecen o desaparecen, esto es: el coste de fijar la línea $p_i, p_{i+1}, \ldots, p_j$ es $(j-i)|b'-b|$, siendo b' el ancho real de los espacios, es decir $(L - l_i - l_{i+1} - ... - l_j)/(j-i)$. No obstante, si $j = k$ (la última línea) el costo será cero a menos que $b' < b$ ya que no es necesario ampliar la última linea. Se pide plantear un algoritmo ávido para resolver el problema, implementarlo y dar un ejemplo donde este algoritmo no encuentre solución maximal o bien demostrar que tal ejemplo no existe.

10. Dado un grafo no dirigido $G(V, E)$, un *conjunto independiente* es un subconjunto de vértices $S \subseteq V$ tal que para todo $v, w \in S$, $(v, w) \notin E$, i.e., no hay vértices en S que estén conectados entre sí. Se desea *obtener el conjunto independiente de tamaño máximo* (con más vértices).

    a) Demostrar que el problema exhibe la propiedad de subestructura óptima.

    b) Confeccionar un algoritmo voraz para resolver el problema. El grafo viene dado por su matriz de adyacencia.

    c) Demostrar que el algoritmo anterior siempre encuentra la solución óptima, o mostrar un contraejemplo en el que no la encuentre.

11. Un fabricante de sombreros de copa recibe n pedidos en su taller. Cada uno de estos incluye una fecha límite de entrega d_i, transcurrida la cual la politica de la empresa obliga a rebajar el precio de venta. Dado que todos los pedidos tienen el mismo precio y requieren el mismo tiempo de confección (1 semana), se desea determinar el orden óptimo de atención de los mismos.

    a) Construir un algoritmo voraz para resolver el problema, e indicar en qué sentido dicho algoritmo es voraz.

    b) Determinar la complejidad del algoritmo diseñado.

    c) Demostrar que el algoritmo proporciona la solución óptima, o encontrar un contraejemplo en el que no la encuentre.

12. Sea un conjunto S de n pueblos, y sea d_{ij} la distancia entre los pueblos i y j. Cualquier pueblo debe tener un hospital a no más de K km de distancia. Se desea determinar en qué pueblos han de abrirse hospitales de acuerdo con la mencionada restricción, de forma que se minimice el número de hospitales necesarios.

    a) Describir una solución voraz al problema, e ilustrarla con un ejemplo que incluya al menos 7 ciudades.

    b) Especificar un algoritmo en pseudocódigo para la estrategia voraz concebida en el apartado anterior.

    c) Demostrar que la estrategia anterior devuelve la solución óptima, o dar un contraejemplo en el que no la encuentre.

13. Una misteriosa enfermedad de origen desconocido se está propagando por una indómita región de Gondwana. Cada pueblo de esta región puede modelarse como un grafo con n vértices cada uno de los cuales representa un habitante de la población, y en el que las aristas representan contacto físico entre individuos. La enfermedad considerada se caracteriza por un patrón de contagio múltiple: sólo si un individuo está conectado con al menos k individuos infectados, pasa a estar infectado él también. Se ha podido desarrollar una vacuna para esta enfermedad, pero es muy costosa por lo que se quiere optimizar su distribución entre la población. El objetivo es, dado un grafo que representa las interacciones entre los individuos de un pueblo, determinar qué individuos deben ser inmunizados de manera que en el caso de que hubiera un brote infeccioso no se produjera contagio.

    a) Definir un algoritmo voraz para resolver el problema, e indicar en qué sentido es voraz.

    b) Construir un ejemplo con $n = 10$ individuos y $k = 3$, e indicar paso a paso cómo se aplicaría el algoritmo anterior.

14. A lo largo de la Autopista de la Costa se han construido numerosas urbanizaciones a las que hay que dar servicio de telefonía móvil. Suponiendo que conocemos los puntos kilométricos $k_1, \cdots, k_n$ ($k_i < k_j$ para $i < j$) en los que hay urbanizaciones, y que la cobertura de una antena de telefonía es de C km, desamos determinar en qué puntos kilométricos (haya en ellos urbanización o no) hay que situar una antena de manera que todas las urbanizaciones tengan cobertura y que el número de antenas sea mínimo.

    a) Definir un algoritmo voraz para resolver el problema, e indicar en qué sentido es voraz.

    b) Construir un ejemplo con $n = 10$ urbanizaciones y $C = 4$ km, e indicar paso a paso cómo se aplicaría el algoritmo anterior.

15. Una regla de Golomb es una regla en la que las marcas no están dispuestas a intervalos regulares de 1mm como en las reglas habituales, sino que las posiciones de éstas satisfacen la siguiente restricción: nunca se repiten las distancias entre dos marcas cualesquiera (es decir, si hay una marca en la posición r y otra en la posición s ($r < s$), no existen marcas en las posiciones i, j ($i < j$) si $s - r = j - i$). Por ejemplo, una regla de Golomb con 4 marcas es $\langle 0, 1, 4, 6\rangle$. Dado un número n de marcas deseado, se desea encontrar la regla de Golomb de longitud mínima (cuya última marca tiene el menor valor).

16. **[2.5p]** Supongamos que en una galería de arte hay $n \geq 1$ pinturas dispuestas en un pasillo de longitud $L > 0$, en las posiciones $\{x_1, \cdots, x_n\}(x_i \in \mathbb{R})$. Supongamos que un vigilante puede asegurar las pinturas que están a una distancia de, a lo sumo, una unidad desde su posición. Diseña un algoritmo voraz que encuentre el mínimo número de vigilantes necesarios para que todos los cuadros estén vigilados de forma segura.