

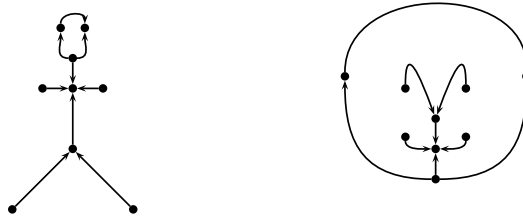
Facultad de Informática  
Metodología y Tecnología de la Programación

Clara Segura, Alberto Verdejo, Marzo 2004

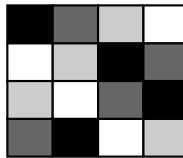
## Vuelta atrás

**Ejercicio 1** La agencia matrimonial Su Media Naranja asegura *matrimonios estables* a sus clientes. Para ello dispone de información almacenada en dos matrices de números naturales  $H[1..n, 1..n]$  y  $M[1..n, 1..n]$  tales que  $H[i, j]$  indica la preferencia de un hombre  $i$  por una mujer  $j$  y  $M[i, j]$  la preferencia de una mujer  $i$  por un hombre  $j$ , para  $i$  y  $j$  entre 1 y  $n$ . Establecida una asignación de  $n$  parejas, si existe algún hombre y alguna mujer que, sin estar emparejados entre sí, se prefieren mutuamente a sus respectivas parejas, entonces la asignación es *inestable*; si no se da tal caso, la asignación es *estable*. Desarrollar un algoritmo que encuentre todos los emparejamientos estables.

**Ejercicio 2** Dos grafos dirigidos  $G$  y  $G'$  son isomorfos si existe una función biyectiva  $f$  entre los vértices de ambos grafos de forma que  $\langle x, y \rangle$  es una arista de  $G$  si y solo si  $\langle f(x), f(y) \rangle$  es una arista de  $G'$ . Por ejemplo, los grafos mostrados a continuación son isomorfos. Diseñar un algoritmo que verifique si dos grafos dados son isomorfos.



**Ejercicio 3** Un *cuadrado latino* de tamaño  $n$  es un tablero de  $n \times n$  posiciones, cada una de las cuales está pintada de un color escogido entre  $n$  diferentes. Ha de cumplir la restricción de que no puede haber colores repetidos en ninguna de sus filas ni en ninguna de sus columnas. La siguiente figura muestra un cuadrado latino de tamaño 4.



Dados  $n$  colores diferentes:

- Escribir un algoritmo que imprima todos los cuadrados latinos posibles de tamaño  $n$ .
- Considerando que una solución es equivalente a todas las que se generen a partir de ella simplemente mediante la permutación de los colores, modificar el algoritmo del apartado anterior para que solo se obtenga un representante de cada clase de equivalencia.

**Ejercicio 4** La República de Fanfanisflán acaba de abrir  $n$  factorías de tecnología punta, que requieren una instalación informática y personal que la dirija. El Gobierno Supremo les ha asignado  $n$  máquinas y otros tantos técnicos recién licenciados, que deberán ser distribuidos entre las fábricas. Como no todas las máquinas son apropiadas para las necesidades de todas las fábricas, se dispone de una función booleana  $\text{apropiada?}(m, f)$  que indica si la máquina  $m$  es apropiada para la fábrica  $f$ . Asimismo, no todos los técnicos conocen todas las máquinas, por lo que se dispone de la correspondiente función  $\text{conoce?}(t, m)$  que indica si el técnico  $t$  conoce la máquina  $m$ . Finalmente, no todos los técnicos están dispuestos a trabajar en todas las fábricas, ya que algunas de ellas están relacionadas con la industria bélica, por lo que se dispone de la función  $\text{aceptaría?}(t, f)$  que indica si el técnico  $t$  estaría dispuesto a trabajar para la fábrica  $f$ . Se pide encontrar, si la hubiera, alguna forma factible de asignar a cada fábrica una máquina apropiada a sus necesidades, y un técnico que conozca dicha máquina y acepte trabajar para la fábrica en cuestión.

**Ejercicio 5** Diseñar un algoritmo que encuentre todas las formas de colocar  $n$  reinas sobre un tablero de ajedrez de tamaño  $n \times n$  de tal forma que no haya dos reinas dándose *jaque*, es decir, que no haya dos reinas en una misma fila, columna o diagonal.

**Ejercicio 6** Dado un tablero de ajedrez de  $n \times n$  posiciones, y un caballo colocado en una posición arbitraria  $\langle x, y \rangle$ , se pide diseñar un algoritmo que determine (si es que existe) una secuencia de  $n^2 - 1$  movimientos de caballo de tal forma que se visiten todos los cuadros del tablero una sola vez. Modificar el algoritmo añadiendo el requisito adicional de que, con un último movimiento, el caballo debe volver a la casilla de partida.

**Ejercicio 7** A partir de una matriz booleana  $L[1..n, 1..n]$  se puede representar un laberinto de la siguiente forma: a partir de una casilla dada, los movimientos posibles son desplazarse a cada una de las cuatro casillas adyacentes (vertical y horizontalmente). Si  $L[i, j] = \text{cierto}$  se puede pasar por la casilla  $\langle i, j \rangle$ , y si  $L[i, j] = \text{falso}$  no se puede pasar por esa casilla. Suponiendo que  $L[1, 1] = L[n, n] = \text{cierto}$ , escribir un algoritmo que encuentre, si existe, un camino de la casilla  $\langle 1, 1 \rangle$  a la casilla  $\langle n, n \rangle$ .

**Ejercicio 8** Sea  $G$  un grafo dirigido de  $n$  nodos, representado por su matriz de adyacencia booleana. Diseña un algoritmo que cuente todas las posibles maneras de colocar los  $n$  nodos respetando el orden topológico inducido por  $G$ ; esto es, si  $a$  y  $b$  son dos nodos distintos y  $G[a, b]$ , entonces  $a$  debe aparecer antes que  $b$ . Dicho de otra forma, la permutación  $x_1, \dots, x_n$ , ( $1 \leq x_i \leq n$ ) es una secuencia válida si y solo si:

$$i < j \rightarrow \neg G[x_j, x_i] \quad 1 \leq i, j \leq n$$

**Ejercicio 9** El Ministro de Desinformación y Decencia se ha propuesto hacer trabajar en firme a sus  $n$  funcionarios, para lo que se ha sacado de la manga  $n$  trabajos. A pesar de su ineficacia, todos los funcionarios son capaces de hacer cualquier trabajo, aunque unos tardan más que otros y unos los hacen peor que otros. La información al respecto se recoge en dos tablas  $T[1..n, 1..n]$  y  $E[1..n, 1..n]$ , donde  $T[i, j]$  representa el tiempo que el funcionario  $i$  tarda en realizar el trabajo  $j$  y  $E[i, j]$  representa la eficacia con la que el funcionario  $i$  realiza el trabajo  $j$ . Para justificar su puesto, Su Excelencia el Sr. Ministro desea conocer la asignación óptima de trabajos a funcionarios en cada uno de los dos sentidos diferentes (e independientes) siguientes:

- (a) de modo que la suma total de tiempos sea mínima, y
- (b) de modo que la suma total de eficacias sea máxima.

**Ejercicio 10** El tío Facundo posee  $n$  huertas, cada una con un tipo diferente de árboles frutales. Las frutas ya han madurado y es hora de recolectarlas. El tío Facundo conoce, para cada una de las huertas, el beneficio  $b_i$  que obtendría por la venta de lo recolectado. El tiempo que se tarda en recoger los frutos de cada finca es asimismo variable y viene dado por  $t_i$ . También sabe los días  $d_i$  que tardan en pudrirse los frutos de cada huerta. Ayudar al tío Facundo a decidir qué debe recolectar para maximizar el beneficio total obtenido.

**Ejercicio 11** 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 12** 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 13** 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 14** 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 15** Se tiene una colección de  $n$  objetos “moldeables”, cada uno con un volumen  $v_i$ , para  $i$  entre 1 y  $n$ , que hay que empaquetar utilizando envases de capacidad  $E$ . Diseñar un algoritmo que calcule el empaquetamiento óptimo, es decir, que minimice la cantidad de envases utilizados, teniendo en cuenta que los objetos no se pueden fraccionar.

**Ejercicio 16** Juanito el Explorador está pasando unas vacaciones en Tombuctú. Juanito disfruta mandando postales de los exóticos lugares que visita, para que sus amigos rabien de envidia cuando las reciban. A tal efecto se ha pasado por la oficina de correos más cercana y se ha hecho con un lote de sellos de  $n$  valores diferentes, disponiendo de *tres* sellos de cada valor. En correos también le han informado de las diferentes tarifas de franqueo de tarjetas postales, y le han explicado que en la esquina superior derecha de cada postal aparece un bloque remarcado destinado a su franqueo. Dicho bloque está dividido en *cinco* casillas, destinadas cada una a un sello, de forma que un franqueo solo es admisible si se alcanza la tarifa correspondiente y se cubren exactamente las cinco casillas.

## Bibliografía

- [1] E. Horowitz, S. Sahni y S. Rajasekaran. *Computer Algorithms*. Computer Science Press, 1998.
- [2] N. Martí, Y. Ortega y A. Verdejo. *Estructuras de datos y métodos algorítmicos. Ejercicios resueltos*. Prentice Hall, 2003.
- [3] R. Neapolitan y K. Naimipour. *Foundations of algorithms*. Jones and Bartlett Publishers, 1997.