



Análisis y Diseño de Algoritmos

Grado en Ingeniería Informática

Problemas del Tema 3

- 1.- A lo largo de un río hay n aldeas. En cada aldea se puede alquilar una canoa que se puede devolver en cualquier otra aldea que esté a favor de corriente (es imposible remar en contra de la corriente). Para todo posible punto de partida i , y para todo posible punto de llegada j , se conoce el coste de un alquiler desde i hasta j . Sin embargo, puede ocurrir que el coste del alquiler desde i hasta j sea mayor que el coste total de una serie de alquileres más breves. En tal caso, se puede devolver la primera canoa en alguna aldea k entre i y j , y seguir camino en una segunda canoa. No hay cargos adicionales por cambiar de canoa de esta manera.

Dar un algoritmo eficiente para determinar el coste mínimo de un viaje en canoa desde todos los posibles puntos de partida i hasta todos los posibles puntos de llegada j . En términos de n , ¿cuánto tiempo requiere su algoritmo?

- 2.- Risk es un juego de mesa en el que varios jugadores tratan de conquistar el mundo. El tablero de juego consiste en un mapa que está dividido en varios países. En el turno de un jugador, los ejércitos que están en un país solamente pueden atacar los países con los que tienen una frontera común. Una vez conquistado el país, los ejércitos se pueden mover al país recién conquistado. Durante el desarrollo del juego, un jugador suele realizar una serie de conquistas con el objetivo de trasladar ejércitos desde un país de origen a un país de destino. Para ello, la mejor estrategia es seleccionar los países de manera que se minimice el número total de países que es necesario conquistar.

El tablero de juego tiene 20 países y entre ellos hay n fronteras, que están almacenadas en la matriz *fronteras*[1.. n][1..2]. Para cada frontera i , se indican los dos países que hacen frontera: *frontera*[i][1] y *frontera*[i][2].

Construya un algoritmo que a partir de un conjunto países de origen y destino, indique la secuencia mínima de países que hay que conquistar. El conjunto de países están almacenados en la matriz *pares*[1.. m][1..2], en la que hay m pares de países. Para cada par de países j , hay que indicar la secuencia mínima de países a conquistar desde el país origen *pares*[j][1] al país destino *pares*[j][2].

- 3.- Juan quiere hacerse un tatuaje en la espalda que conecte todas las pecas que tiene. La idea es conectar las pecas empleando líneas rectas entre pares de pecas, de manera que debe existir una secuencia de líneas que conecten cualquier peca a cualquier otra peca. El coste del tatuaje será proporcional a la suma de la longitud de todas la líneas dibujadas. Construya un algoritmo que obtenga el tatuaje de coste mínimo y que imprima en pantalla la suma de la longitud de todas la líneas del mismo.

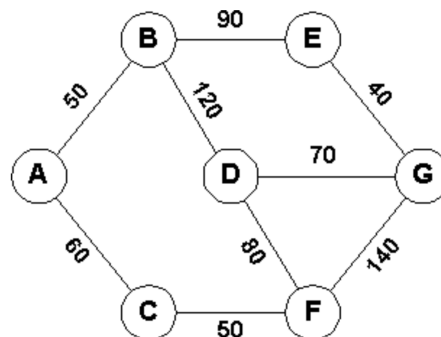
La espalda de Juan tiene n pecas, cuyas coordenadas se proporcionan en la matriz *pecas*[1.. n][1..2]. Para cada peca i , sus coordenadas (x_i, y_i) son: $x_i = \text{pecas}[i][1]$ y $y_i = \text{pecas}[i][2]$. Las coordenadas de las pecas son números reales.

- 4.- Juan tiene que realizar n tareas, numeradas de 1 a n . Las tareas no son independientes, y la ejecución de una tarea solamente es posible si otras tareas ya han sido ejecutadas. Existen m relaciones de dependencia entre tareas, que se proporcionan en la matriz *depende*[1.. m][1..2]. Cada relación de dependencia i indica que la tarea *depende*[i][1] debe ser ejecutada antes que la tarea *depende*[i][2].

Construya un algoritmo que muestre por pantalla el orden en el que se tienen que ejecutar las tareas para que se cumplan todas las relaciones de dependencia.



- 5.- El grado en Ingeniería Informática por la Universidad de Rivia tienen M asignaturas, todas ellas obligatorias. Cada asignatura tiene 0 o más asignaturas requisito. En total hay N requisitos, que están almacenados en la matriz $prerequisitos[1..N][1..2]$. Para cada requisito i , se indica que es preciso cursar la asignatura $prerequisitos[i][1]$ para poder cursar la asignatura $prerequisitos[i][2]$. Todas las asignaturas son cuatrimestrales y los cuatrimestres se numeran consecutivamente: 1, 2, 3, ... Diseñe un algoritmo que muestre por pantalla el menor cuatrimestre en el que se puede cursar cada asignatura del grado.
- 6.- Nos proporcionan una matriz $M[1..n, 1..n]$ que representa un laberinto. En cada posición de la matriz un 0 representa pasillo y un 1 representa muro. Los movimientos permitidos son: izquierda, derecha, arriba y abajo. La entrada del laberinto está en la posición $(1, e)$ y la salida en la posición (n, s) . Desarrolle un algoritmo que imprima por pantalla la manera de llegar desde la entrada del laberinto hasta la salida. Si no es posible llegar desde la entrada hasta la salida, debe imprimir "No hay solución".
- 7.- En este problema vamos a considerar la contaminación acústica de una ciudad. Considere el siguiente mapa de la ciudad donde las aristas se refieren a calles y los nodos se refieren a cruces. El número entero en cada arista es el nivel de intensidad promedio del sonido (en decibelios) en la calle correspondiente.



Para ir del cruce A al cruce G, se puede seguir el siguiente camino: A-C-F-G. En ese caso, hay que ser capaz de tolerar una intensidad de sonido tan alta como 140 decibelios. Para los caminos A-B-E-G, A-B-D-G y A-C-F-D-G hay que tolerar respectivamente 90, 120 y 80 decibelios de intensidad de sonido. También hay otros caminos. Sin embargo, está claro que A-C-F-D-G es el camino más cómodo, ya que no te exige tolerar más de 80 decibelios.

En este problema, dado un mapa de la ciudad, debe determinar cuál es el camino más cómodo para ir de un cruce determinado a otro. El camino más cómodo es aquel en el que el nivel de intensidad de sonido que hay que ser capaz de tolerar es el mínimo posible. Desarrolle un algoritmo que imprima por pantalla el camino más cómodo para ir del cruce x al cruce y . Si no existe camino debe imprimir "no hay camino".