Estructuras de Datos no Lineales

Práctica 7

Problemas de grafos II

TRABAJO PREVIO

Antes de asistir a la sesión de prácticas es obligatorio:

1. Imprimir copia de este enunciado.
2. Lectura profunda del mismo.
3. Reflexión sobre el contenido de la práctica y generación de la lista de dudas asociada a dicha práctica y a los problemas que la componen.
4. **Esbozo serio de solución** de los problemas en papel (al menos de los que se hayan entendido).

PASOS A SEGUIR

1. Para cada uno de los problemas escribir un módulo (de nombre por ejemplo `ejercicioN.cpp`) que contenga las funciones requeridas en el enunciado, para lo cual se hará uso de las clases y algoritmos de grafos proporcionados.
2. Escribir un programa de prueba de la solución propuesta para el problema, donde se realicen las llamadas a las funciones correspondientes definidas en el paso anterior, comprobando el resultado de salida para una batería suficientemente amplia de casos de prueba. Esto se puede hacer de dos maneras: Incluyendo la función `main()` en el fichero `ejercicioN.cpp` del paso anterior; o bien, creando un nuevo fichero `.cpp` para la función `main()`, que se compilará por separado y se enlazará con este `ejercicioN.cpp` anterior.

PROBLEMAS

1. Tu agencia de viajes "OTRAVEZUNGRAFO S.A." se enfrenta a un curioso cliente. Es un personaje sorprendente, no le importa el dinero y quiere hacer el viaje más caro posible entre las ciudades que ofertas. Su objetivo es gastarse la mayor cantidad de dinero posible (ojalá todos los clientes fueran así), no le importa el origen ni el destino del viaje.

Sabiendo que es imposible pasar dos veces por la misma ciudad, ya que casualmente el grafo de tu agencia de viajes resultó ser acíclico, devolver el coste, origen y destino de tan curioso viaje. Se parte de la matriz de costes directos entre las ciudades del grafo.

2. Se dispone de un laberinto de *NxN* casillas del que se conocen las casillas de entrada y salida del mismo. Si te encuentras en una casilla sólo puedes moverte en las siguientes cuatro direcciones (arriba, abajo, derecha, izquierda). Por otra parte, entre algunas de las

casillas hay una pared que impide moverse entre las dos casillas que separa dicha pared (en caso contrario no sería un verdadero laberinto).

Implementa un subprograma que dados

- N (dimensión del laberinto),
- la lista de paredes del laberinto,
- la casilla de entrada, y
- la casilla de salida,

calcule el camino más corto para ir de la entrada a la salida y su longitud.

3. Eres el orgulloso dueño de una empresa de distribución. Tu misión radica en distribuir todo tu stock entre las diferentes ciudades en las que tu empresa dispone de almacén.

Tienes un grafo representado mediante la matriz de costes, en el que aparece el coste (por unidad de producto) de transportar los productos entre las diferentes ciudades del grafo.

Pero además resulta que los Ayuntamientos de las diferentes ciudades en las que tienes almacén están muy interesados en que almacenes tus productos en ellas, por lo que están dispuestos a subvencionarte con un porcentaje de los gastos mínimos de transporte hasta la ciudad. Para facilitar el problema, consideraremos despreciables los costes de volver el camión a su base (centro de producción).

He aquí tu problema. Dispones de

- el centro de producción, nodo origen en el que tienes tu producto (no tiene almacén),
- una cantidad de unidades de producto (*cantidad*),
- la matriz de costes del grafo de distribución con *N* ciudades,
- la capacidad de almacenamiento de cada una de ellas,
- el porcentaje de subvención (sobre los gastos mínimos) que te ofrece cada Ayuntamiento.

Las diferentes ciudades (almacenes) pueden tener distinta capacidad, y además la capacidad total puede ser superior a la *cantidad* disponible de producto, por lo que debes decidir cuántas unidades de producto almacenas en cada una de las ciudades.

Debes tener en cuenta además las subvenciones que recibirás de los diferentes Ayuntamientos, las cuales pueden ser distintas en cada uno y estarán entre el 0% y el 100% de los costes mínimos.

La solución del problema debe incluir las cantidades a almacenar en cada ciudad bajo estas condiciones y el coste mínimo total de la operación de distribución para tu empresa.

4. Eres el orgulloso dueño de la empresa “Cementos de Zuelandia S.A”. Empresa dedicada a la fabricación y distribución de cemento, sita en la capital de Zuelandia. Para la distribución del cemento entre tus diferentes clientes (ciudades de Zuelandia) dispones de una flota de camiones y de una plantilla de conductores zuelandeses.

El problema a resolver tiene que ver con el carácter del zuelandés. El zuelandés es una persona que se toma demasiadas “libertades” en su trabajo, de hecho, tienes fundadas sospechas de que tus conductores utilizan los camiones de la empresa para usos particulares (es decir indebidos, y a tu costa) por lo que quieres controlar los kilómetros que recorren tus camiones.

Todos los días se genera el parte de trabajo, en el que se incluyen el número de cargas de cemento (1 carga = 1 camión lleno de cemento) que debes enviar a cada cliente (cliente = ciudad de Zuelandia). Es innecesario indicar que no todos los días hay que enviar cargas a todos los clientes, y además, puedes suponer razonablemente que tu flota de camiones es capaz de hacer el trabajo diario.

Para la resolución del problema quizá sea interesante recordar que Zuelandia es un país cuya especial orografía sólo permite que las carreteras tengan un sentido de circulación.

Implementa una función que dado el grafo con las distancias directas entre las diferentes ciudades zuelandesas, el parte de trabajo diario, y la capital de Zuelandia, devuelva la distancia total en kilómetros que deben recorrer tus camiones en el día, para que puedas descubrir si es cierto o no que usan tus camiones en actividades ajenas a la empresa.

Nota Importante:

A partir del problema 5 (el viajero alérgico), empiezan a aparecer en los enunciados el uso de diferentes medios de transporte a la hora de realizar un viaje. En nuestros problemas (tanto en prácticas como en exámenes) asumiremos que

a) Definición de trasbordo : En el contexto de los problemas de la asignatura, consideraremos trasbordo el cambio de medio de transporte**.**

b) Trasbordos libres y gratuitos por defecto: Si el enunciado del problema no indica lo contrario los trasbordos en nuestros problemas son libres y gratuitos**.**

5. Se dispone de tres grafos que representan la matriz de costes para viajes en un determinado país pero por diferentes medios de transporte, por supuesto todos los grafos tendrán el mismo número de nodos. El primer grafo representa los costes de ir por carretera, el segundo en tren y el tercero en avión. Dado un viajero que dispone de una determinada cantidad de dinero, que es alérgico a uno de los tres medios de transporte, y que sale de una ciudad determinada, implementar un subprograma que determine las ciudades a las que podría llegar nuestro infatigable viajero.

6. Al dueño de una agencia de transportes se le plantea la siguiente situación. La agencia de viajes ofrece distintas trayectorias combinadas entre *N* ciudades españolas utilizando tren y autobús. Se dispone de dos grafos que representan los costes (matriz de costes) de viajar entre diferentes ciudades, por un lado en tren, y por otro en autobús (por supuesto entre las ciudades que tengan línea directa entre ellas). Además coincide que los taxis de toda España se encuentran en estos momentos en huelga general, lo que implica que sólo se podrá cambiar de transporte en una ciudad determinada en la que, por casualidad, las estaciones de tren y autobús están unidas.

Implementa una función que calcule la tarifa mínima (matriz de costes mínimos) de viajar entre cualesquiera de las *N* ciudades disponiendo del grafo de costes en autobús, del grafo de costes en tren, y de la ciudad que tiene las estaciones unidas.

7. Se dispone de dos grafos (matriz de costes) que representan los costes de viajar entre *N* ciudades españolas utilizando el tren (primer grafo) y el autobús (segundo grafo). Ambos grafos representan viajes entre las mismas *N* ciudades.

Nuestro objetivo es hallar el camino de coste mínimo para viajar entre dos ciudades concretas del grafo, *origen* y *destino*, en las siguientes condiciones:

- La ciudad *origen* sólo dispone de transporte por tren.
- La ciudad *destino* sólo dispone de transporte por autobús.
- El sector del taxi, bastante conflictivo en nuestros problemas, sigue en huelga, por lo que únicamente es posible cambiar de transporte en dos ciudades del grafo, *cambio1* y *cambio2*, donde las estaciones de tren y autobús están unidas.

Implementa un subprograma que calcule la ruta y el coste mínimo para viajar entre las ciudades *Origen* y *Destino* en estas condiciones.

8. "UN SOLO TRANSBORDO, POR FAVOR". Este es el título que reza en tu flamante compañía de viajes. Tu publicidad explica, por supuesto, que ofreces viajes combinados de TREN y/o AUTOBÚS (es decir, viajes en tren, en autobús, o usando ambos), entre *N* ciudades del país, que ofreces un servicio inmejorable, precios muy competitivos, y que garantizas ante notario algo que no ofrece ninguno de tus competidores: que en todos tus viajes COMO MÁXIMO se hará un solo transbordo (cambio de medio de transporte).

Bien, hoy es 1 de Julio y comienza la temporada de viajes.

¡Qué suerte! Acaba de aparecer un cliente en tu oficina. Te explica que quiere viajar entre dos ciudades, *Origen* y *Destino*, y quiere saber cuánto le costará.

Para responder a esa pregunta dispones de dos grafos de costes directos (matriz de costes) de viajar entre las *N* ciudades del país, un grafo con los costes de viajar en tren y otro en autobús.

Implementa un subprograma que calcule la tarifa mínima en estas condiciones.

Mucha suerte en el negocio, que la competencia es dura.

9. Se dispone de dos grafos que representan la matriz de costes para viajes en un determinado país, pero por diferentes medios de transporte (tren y autobús, por ejemplo). Por supuesto ambos grafos tendrán el mismo número de nodos, *N*. Dados ambos grafos, una ciudad de origen, una ciudad de destino y el coste del taxi para cambiar de una estación a otra dentro de cualquier ciudad (se supone constante e igual para todas las ciudades), implementa un subprograma que calcule el camino y el coste mínimo para ir de la ciudad origen a la ciudad destino.

10. Se dispone de tres grafos que representan la matriz de costes para viajes en un determinado país, pero por diferentes medios de transporte (tren, autobús y avión). Por supuesto los tres grafos tendrán el mismo número de nodos, *N*.

Dados los siguientes datos:

- los tres grafos,
- una ciudad de origen,
- una ciudad de destino,
- el coste del taxi para cambiar, dentro de una ciudad, de la estación de tren a la de autobús o viceversa (*taxi-tren-bus*) y
- el coste del taxi desde el aeropuerto a la estación de tren o la de autobús, o viceversa (*taxi-aeropuerto-tren/bus*)

y asumiendo que ambos costes de taxi (distintos entre sí, son dos costes diferentes) son constantes e iguales para todas las ciudades, implementa un subprograma que calcule el camino y el coste mínimo para ir de la ciudad origen a la ciudad destino.

11. Disponemos de tres grafos (matriz de costes) que representan los costes directos de viajar entre las ciudades de tres de las islas del archipiélago de las Huríes (Zuelandia). Para poder viajar de una isla a otra se dispone de una serie de puentes que conectan ciudades de las diferentes islas a un precio francamente asequible (por decisión del Prefecto de las Huríes, el uso de los puentes es absolutamente gratuito).

Si el alumno desea simplificar el problema, puede numerar las N_1 ciudades de la isla 1, del 0 al N_1-1, las N_2 ciudades de la isla 2, del N_1 al N_1+N_2-1, y las N_3 de la última, del N_1+N_2 al $N_1+N_2+N_3-1$.

Disponiendo de las tres matrices de costes directos de viajar dentro de cada una de las islas, y la lista de puentes entre ciudades de las mismas, calculad los costes mínimos de viajar entre cualesquiera dos ciudades de estas tres islas.

¡¡¡ QUE DISFRUTÉIS EL VIAJE !!!

12. El archipiélago de Grecoland (Zuelandia) está formado únicamente por dos islas, Fobos y Deimos, que tienen N_1 y N_2 ciudades, respectivamente, de las cuales C_1 y C_2 ciudades son costeras (obviamente $C_1 \leq N_1$ y $C_2 \leq N_2$). Se desea construir un puente que una ambas islas. Nuestro problema es elegir el puente a construir entre todos los posibles, sabiendo que el coste de construcción del puente se considera irrelevante. Por tanto, escogeremos aquel puente que minimice el coste global de viajar entre todas las ciudades de las dos islas, teniendo en cuenta las siguientes premisas:

1. Se asume que el coste viajar entre las dos ciudades que una el puente es 0.
2. Para poder plantearse las mejoras en el transporte que implica la construcción de un puente frente a cualquier otro, se asume que se realizarán exactamente el mismo número de viajes entre cualesquiera ciudades del archipiélago. Por ejemplo, se considerará que el número de viajes entre la ciudad P de Fobos y la Q de Deimos será el mismo que entre las ciudades R y S de la misma isla. Dicho de otra forma, todos los posibles trayectos a realizar dentro del archipiélago son igual de importantes.

Dadas las matrices de costes directos de Fobos y Deimos y las listas de ciudades costeras de ambas islas, implementa un subprograma que calcule las dos ciudades que unirá el puente.

13. El archipiélago de las Huríes acaba de ser devastado por un maremoto de dimensiones desconocidas hasta la fecha. La primera consecuencia ha sido que todos y cada uno de los puentes que unían las diferentes ciudades de las tres islas han sido destruidos. En misión de urgencia las Naciones Unidas han decidido construir el mínimo número de puentes que permitan unir las tres islas. Asumiendo que el coste de construcción de los puentes implicados los pagará la ONU, por lo que se considera irrelevante, nuestro problema es decidir qué puentes deben construirse. Las tres islas de las Huríes tienen respectivamente N_1, N_2 y N_3 ciudades, de las cuales C_1, C_2 y C_3 son costeras (obviamente $C_1 \leq N_1$, $C_2 \leq N_2$ y $C_3 \leq N_3$) . Nuestro problema es elegir los puentes a construir entre todos los posibles. Por tanto, escogeremos aquellos puentes que minimicen el coste global de viajar entre todas las ciudades de las tres islas, teniendo en cuenta las siguientes premisas:

1. Se asume que el coste viajar entre las ciudades que unan los puentes es 0.
2. La ONU subvencionará únicamente el número mínimo de puentes necesario para comunicar las tres islas.
3. Para poder plantearse las mejoras en el transporte que implica la construcción de un puente frente a cualquier otro, se asume que se realizarán exactamente el mismo número de viajes entre cualesquiera ciudades del archipélago. Dicho de

otra forma, todos los posibles trayectos a realizar dentro del archipiélago son igual de importantes.

Dadas las matrices de costes directos de las tres islas y las listas de ciudades costeras del archipiélago, implementad un subprograma que calcule los puentes a construir en las condiciones anteriormente descritas.

1- Reescribo floyd para que calcule el máximo coste para llegar a un punto, hay que tener en cuenta el infinito

¡¡ojo!!

Si inviertes los tCoste de los caminos (multiplico $\cdot(-1)$) al hacer floyd devolvera los más caro, luego dada la matriz tengo que invertir la matriz para que no haya num negativo

2- Crear un grafo dado el tamaño, tendré que usar la matriz de adyacencia. Una vez creado tengo en cuenta las paredes, usar la funcion camino que viene en el tad. Para obtener la coordeanda divido el num del nodo entre m y obtengo x, el resto será y. La complejidad reside en construir el grafo ya que es muy tipico de examenes. Si hago dijkstra tngo el camino de origen a destino y sus pasos. La lista de apreces es un vector de pares de vertices rollo de x1,y,1 a x2,y2, significa que esos no se peuden, se puede hacer una estructura pared que te diga los valores desde donde hasta dodne va la pared, la paredes son de tamaño max 1, por lo que una pared larga será concatenaciones de varais paredes.
Devuelvo el vector de vertices

SI HAGO LA TRASPUESTA A LA MATRIZ Y APLICO DIJKSTRA OBTENGO DIJSTRA INVERSO

3- Calculo los costes minimos a nodos (contando las subvenciones), y compruebo que es mas barato si ir a un almacen donde puedo guardar todo o voy mirando el primer coste minimo y lleno ese almacen, si me queda productos voy al siguiente y así sucesivamete hasta vaciarlo.
Usaría un dijkstra

4- Tengo un vector con la ctdad a repartir a cada nodo i, hago dijkstra y dijkstra inverso para calcular los costes que sera dikstra a nodo i + dijkstra inverso hacia nodo i * ctdad de viajes necesarios a i

5- Hay que ver en cada matriz los costes minimos para llegar a los sitios, usando dijkstra ya que partes de un origen. Este tio o va a el sitio usando un medio o usa otro, no hace transbordo. Hay que hacer dijkstra para los dos tranportes que usa y cuales les llega con el dinero q tiene. Valdría con un vector de booleanos diciendo que puede y ya

6 Se crea un 'grafo en 3d' que conecte el tren con el bus a traves de las estaciones y hago floyd

7 Intuyo que será como el 6 pero solo se permiten dos saltos y se haría Dijkstra. No sé como limitar los transbordo

8 Mismo caso que el 6 pero permitiendo solo una vez saltar de grafo

9 Construyo un grafo de tres alturas y subir o bajar en el coste del taxi y hago dijkstra

10 Construyo un grafo tridimensional con 3 alturas y subir o bajar es el coste del taxi a ese transporte en esa ciudad (nodo) y hago dijkstra

11 Conecto cada isla (grafo) a traves de las ciudades (nodos) que tienen puentes con otra isla y hago Floyd

12 No sé como hacerlo, podría ser hacer dijkstra inverso a todas las ciudades costeras que tengan una media de caminos mas baja y coges la mas baja en cada isla y en esa haces el puente?

13 Misma historia que el 12 pero lo hago para interconectar las tres islas.

---

6 Hago floyd en tren y bus

Para obtener la de los cambios

c1 = Floyd(tren, v1)

c2 = Floyd(Bus, v2)

for (i:0 -> n)

for (j:0 -> n)

-> M[i][j] = min(min(c1[i][j], c2[i][j]),

min(suma(c1[i][cambio], c2[cambio, j]),

suma(c2[i][cambio], c1[cambio, j])))

Se pide como tal el coste mínimo de llegar a los sitios y solo se cambia en una ciudad por lo que hago el minimo de A->B en bus, en tren, en bus -> cambio y tren y al revés

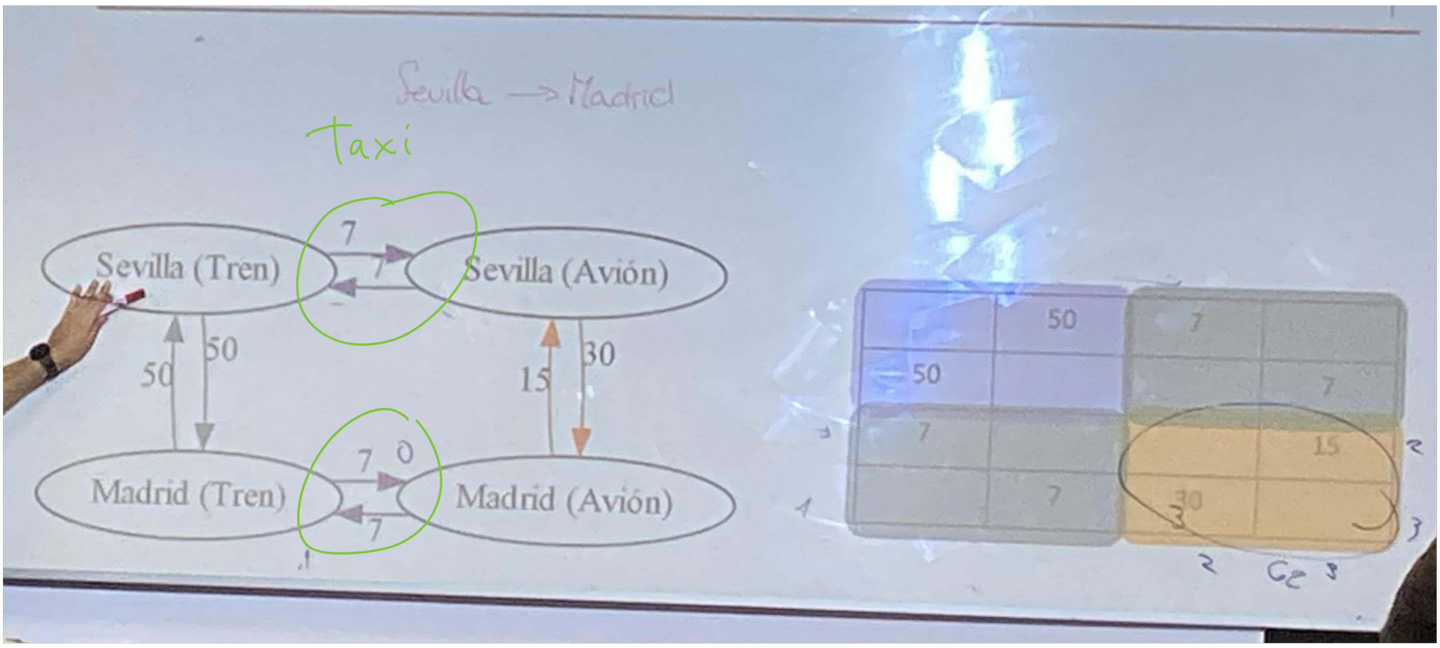
7 Hago dijkstra en el origen para ver los costes de llegar a los cambios y luego dijkstra inverso en el destino, y me quedo con el camino completo que tenga coste más barato tomando un cambio.

Tengo que devolver el camino y coste

↳ vector de vertices

8 Hago dijkstra e inverso con origen y destino (4 dijkstra total) y despues hago el minimo de ir directo en un transporte o la suma de origen a cambio y desde ahi a destino cambio es ir iterando entre todos los nodos y lo mismo con el otro orden

9 Hay que crear un grafo que una los dos transportes para ello creo una matriz de tamaño n + m y la relleno asignando a la matriz su vertice + el num de vertices del otro grafo



10 Parecido al anterior pero hay que tener en cuenta los taxis
Hago 3 dijkstra para obtener los caminos mínimos (cada dijkstra representa que INICIO de origen con cada transporte

11 Tener en cuenta que cada grafo tiene distinto tamaño. Pero mismo caso que el anterior hago un super grafo teniendo en cuenta el tamaño de las islas, pongo un puente entre los nodos que pueda y hago Floyd

12 Hago floyd con todas las posibles construcciones de puente y la mejor construcción será aquella que la suma de Floyd de la matriz entera sea mas barata.

13 El minimo de puentes será v - 1 puentes