Algorísmia. Problemes 4

QT Curs 2017-2018

61. L'arbitratge de divises és una situació en la qual un operador de monedes intel·ligent pot executar una seqüència de canvis de moneda per tal de obtenir una quantitat potencialment il·limitada de diners. Per exemple, suposem que els dòlars nord-americans s'estan comprant en el mercat de divises per 50 rupies i una altra moneda al mercat de divises ven dòlars nord-americans per 40 rupies. En aquesta situació, un operador podria intercanviar 1 milió de dòlars per l'equivalent a 50 milions de rupies i després intercanvien les rupies per 50 milions / 40 = 1,25 milions de dòlars. Les situacions d'arbitratge de divises que ens plantegem són més complexes i implican diversos passos de conversió entre moltes monedes. Per exemple, en el següent quadre de conversió teniu un arbitratge de tres passos.

Problem. Given table of exchange rates, is there an arbitrage opportunity?

	USD	EUR	GBP	CHF	CAD
USD	1	0.741	0.657	1.061	1.011
EUR	1.350	1	0.888	1.433	1.366
GBP	1.521	1.126	1	1.614	1.538
CHF	0.943	0.698	0.620	1	0.953
CAD	0.995	0.732	0.650	1.049	1

Ex. \$1,000 $\Rightarrow$ 741 Euros $\Rightarrow$ 1,012.206 Canadian dollars $\Rightarrow$ \$1,007.14497.

$1000 \times 0.741 \times 1.366 \times 0.995 = 1007.14497$

Dissenyeu un algorisme que, donada una taula de conversió, trobi un arbitratge que ens permeti incrementar, si és posible, la nostra quantitat inicial de diners.

62. En aquest problema, estudiem la relació entre *arbres d'expansió mínims* (MST) i *arbres de camins mínims* en un graf no dirigit G. Recordeu que donat un $G = (V, E)$ amb pesos $w : E \to \mathbb{R}$ i un punt $s \in V$ l'arbre de camins mínims arrelat a s és un subgraf $T' = (V', E')$ de G tal que:

   (a) T' és un arbre, i per tant $|E'| = |V'| - 1$,

   (b) hi a un camí de s fins a qualsevol vertex a V',

   (c) per a qualsevol $u \in V'$, la distancia de s a u a T' és la mateixa que la distancia de s a u a G.

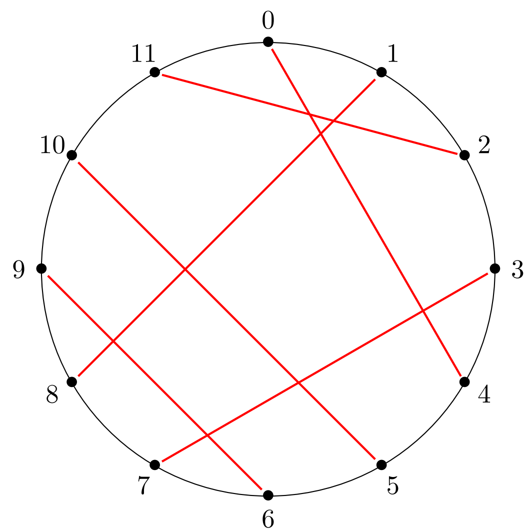
   Recordeu que, igual que succeix amb el MST, donat un $s \in V'$, G pot tenir més d'un arbre de camíns mínims arrelat a s.

   (a) Demostreu si és cert o no que donat qualsevol graf connex i no dirigit G, amb $w : E \to \mathbb{R}^+$, sempre hi ha un arbre de camins mínims T' tal que T' també és un arbre d'expansió mínima a G.

   (b) Demostreu si pot haver-hi un graf no dirigit G amb $w : E \to \mathbb{R}^+$ i connex, tal que G tingui un arbre de camins mínims T' i un MST T que no comparteixen cap aresta.

63. Donat un conjunt de n cordes en el cercle unitat diem que un subconjunt de cordes es *viable* si no hi han dues cordes que es tallen. Volem trobar un subconjunt viable amb mida màxima.

Per resoldre el problema assumim que mai dues cordes tenen un extrem en comú. Per això podem enumerar els extrems de les n cordes de 0 a $2n-1$ seguint el sentit de les agulles del rellotge. Aleshores, l'entrada del problema consisteix en una seqüència de n parelles dels nombres $0, \ldots 2n-1$ on cada i, $0 \leq i \leq 2n-1$, apareix exactament en una parella. La parella (i,j) representa la corda amb extrems i i j. A la figura següent teniu un exemple de instancia amb 6 cordes:



l'entrada corresponent és $(0,4),(1,8)(11,2),(3,7),(5,10),(9,6)$.

Per $0 \leq i < j \leq 2n-1$, definim $T(i,j)$ com la mida del subconjunt viable més gran que es pot formar amb el conjunt de les cordes (a,b) tals que $i \leq a, b \leq j$.

(a) Per $0 \leq i < j \leq 2n-1$, proporcioneu una recurrència que permeti calcular $T(i,j)$.

(b) Proporcioneu un algorisme que, donat un conjunt de cordes en el cercle unitat, obtingui un conjunt viable amb mida màxima en temps polinòmic.

64. El dia de Sant Tomàs, la coral dels alumnes de la facultat organitzen una sortida col·lectiva a esquiar. A l'estació d'esquí on van, tenen, m parells d'esquís, on l'alçada del parell i-èsim és s_i. Sigui n el nombre d'alumnes que no tenen esquís propis i per tant han de llogar-los a l'estació d'esquí. Sigui h_i l'alçada de l'alumne i-ésim (d'entre els alumnes que no tenen esquís). Es ben coneguda la regla que cada esquiador ha d'esquíar amb uns esquís de manera que l'alçada dels esquís sigui el més semblant possible a la seva pròpia alçada. Dissenyeu un algorisme eficient per assignar esquís als n alumnes sense esquís, de manera que es minimitze la diferència total d'alçada entre els esquís i els corresponents alumnes.

65. Les xarxes ad-hoc, composades per dispositius sense fils de baixa potència, s'han proposat per situacions com els desastres naturals en què els coordinadors dels treballs de rescat podrien controlar les condicions en zones de difícil accés. La idea és que una gran col·lecció d'aquests dispositius sense fils es podria llançar des d'un avió en una regió per a continuació reconfigurar-se com una xarxa operativa.

Estem parlant de: (a) dispositius relativament barats, el quals (b) es llancen des d'un avió a (c) un territori perillós; i per la combinació de (a), (b) i (c), es fa necessari fer front a la fallida d'un nombre raonable dels dispositius.

Ens agradaria que fos el cas que si un dels dispositius v detecta que està en perill de fallar, transmetés una representació del seu estat a un altre dispositiu a la xarxa. Cada dispositiu té un abast de transmissió limitat, pot comunicar-se amb altres dispositius que es troben com a màxim a d metres d'ell. Com que no volem que es transmeti el seu estat a un dispositiu inoperant, hem d'incloure una mica de redundància: Un dispositiu v ha de tenir un conjunt de k altres dispositius en un radi de d metres de distància. Anomenarem a aquest conjunt una *còpia de seguretat* per al dispositiu v.

Suposeu que desprès del llançament podem obtenir les coordenades $p_i = (x_i, y_i)$ dels n dispositius operatius que formen la xarxa inicial.

Dissenyeu un algorisme per determinar si és possible triar una còpia de seguretat per a cada dispositiu, amb la propietat addicional que, per algun paràmetre donat b, cap dispositiu apareix en la còpia de seguretat de més de b altres dispositius. L'algorisme ha de proporcionar com a sortida també els conjunts de còpia de seguretat, sempre que es puguin trobar.

Una solució: Resolveremos el problema como un problema de flujo máximo. Para ello construimos una red formado por un grafo $G = (V, E)$ donde $V = \{s, t, u_1, \dots, u_n, v_1, \dots, v_n\}$ y lo arcos en E y sus capaciades son las sigientes:

- Para $i \in [n]$, (s, u_i) con capacidad k y arco (v_i, t) con capacidad b.
- Para $i, \in [n]$ con $i \neq j$ y $d(p_i, p_j) \leq d$, (u_i, u_j) con capacidad 1.

Supongamos que podemos seleccionar copias de seguridad C_i para todo $i \in [n]$ verificando las restricciones. En este caso, par $j \in C_i$ sabemos que $d(p_i, p_j) \leq d$ y por tanto $(u_i, v_j) \in E$. Definimos la siguiente asignación de flujo: para $i \in [n]$ $f(s, u_i) = k$ y para $(u_i, v_j) \in E$, $f(u_i, v_j) = 1$ si $j \in C_i$, $f(u_i, v_j) = 0$ si $j \in C_i$; para $j \in [n]$, $f(u_j, t) = \sum_{i \in [n] i \neq j} f(u_i, v_j)$. Al cumplirse las restricciones, la asignación f es un flujo válido con valor kn.

Supongamos que en la red tenemos un flujo válido f con valor kn, definimos para $i \in [n]$ el conjunto $C_i = \{j \mid (u_i, v_j) \in E \text{ and } f(u_i, v_j) = 1\}$. Como el valor del flujo es kn, cada vértice v_i recibe k unidades de flujo, por la ley de conservación del flujo, tenemos que $|C_i| = k$. Para la segunda restriccion, observemos que $|\{i \mid j \in C_i\}|$ coincide con el flujo de entrada en v_j, de nuevo por la ley de conservación de flujo esta cantidad tiene que coincidir con $f(v_j, t)$. Finalmente al ser un flujo válido tenemos $|\{i \mid j \in C_i\}| = f(v_j, t) \leq c(v_j, t) = b$ y se cumple la segunda restricción.

Por lo tanto tenemos que el problema planteado tiene solución si y solo si el valor del flujo máximo en la red construida es kn.

Por tanto para resolver el problema calcularemos un flujo con valor máximo utilizando un algoritmo que permita obtaner soluciones enteras. Utilizando Ford-Fulkerson el número máximo de augmentations es kn y podemos resolver el problema en $O(kn(|V| + |E|)$ es decir $O(kn^3)$.

Una vez obtenido este flujo f, para $i \in [n]$ obtenemos el conjunto $C_i = \{j \mid (u_i, v_j) \in E \text{ and } f(u_i, v_j) = 1\}$ con un recorrido de los arcos de G en $O(n^2)$ time.

66. En previsió a un increment gran d'accidents de trafic per començament de vacances, un hospital vol avaluar la seva disponibilitat de reserves de sang. Per a fer una transfusió de sang s'ha de tenir cura del tipus de sang que injectem a cada persona. Cada humà té un tipus particular de sang, que pot ser incompatible amb el tipus de sang d'altres persones Hi ha quatre tipus de sang, corresponents als antígens que té la sang. La regla bàsica per a la transfusió de sang és que una persona no pot rebre sang amb un antigen en particular si la seva pròpia sang no conté aquest antigen. Per evitat transfusions mortals, la sang humana es divideix en en quatre tipus: A, B, AB i O. La sang de tipus A té l'antigen A, sang de tipus B té l'antigen B, sang de tipus AB té els dos, i la sang de tipus O no té cap. Això implica que a una transfusió, els pacients amb el tipus A poden rebre únicament els tipus de sang A o O, els pacients amb el tipus B poden rebre només sang del tipus B o O, els pacients amb tipus O poden rebre només O i pacients amb el tipus AB poden rebre de qualsevol dels quatre tipus[1] La quantitat de sang que un centre mèdic disposa es medeix en unitats (borses que contenen 500ml, de sang) i que es el que normalment es posa en una transfusió (a no ser el pacient hagi perdut molta sang). Sigui S_O, S_A, S_B i S_{AB} el subministrament en unitats dels diferent tipus de sang que l'hospital disposa abans del començament de les vacances. S'ha fet una estimació de la quantitat de sang que es necessitara, basant-se en una previsió del nombre de ferits que hi hauran i en la distribució típica dels quatre tipus de sang entre la població.

   (a) Doneu un algoritme polinòmic per a determinar si la sang de que es disposa seria suficient per la necessitat projectada.

   (b) Apliqueu el vostre algorisme al següent exemple: durant la primera setmana de començament de vacances, s'espera es necessiten a com a màxim 100 unitats de sang. La distribució típica dels tipus de sang en pacients es la següent: de tipus O 45 %, de tipus A 42%, de tipus B 10%, i del tipus AB 3 %. L'hospital vol saber si amb les 105 unitats de sang que tenen, seria suficient si 100 pacients arriben amb una distribució esperada dels tipus de sang donada per la següent taula:

Tipus sang	existències	demanda
O	50	45
A	36	42
B	11	10
AB	8	3

   Justifiqueu si amb les 105 unitats de sang disponibles, hi ha suficient per a satisfer les 100 unitats de demanda. Trobeu una assignació que doni servei al màxim nombre possible de pacients. (Utilitzeu un argument basat en el tall mínim d'un graf, per a demostrar què no tots els pacients poden rebre sang).

[1] el problema és una versió simplificada de la realitat, ja que també s'hauria de considerar el factor Rh de la sang

67. Considereu el següent escenari. Ha succeït una catàstrofe a una ciutat gran, el equips de rescat tenen identificats n ferits greus a diferents llocs de la ciutat, que necessiten ser traslladats urgentment a un hospital. Hi han k hospitals disponibles. Degut a la gravetat de les ferides, es important que cada ferit arribe a un hospital abans de 30 minuts. Depenent de a on el ferit esta situat necessita anar-hi a un hospital que estigui com a màxim a 30 minuts de distancia per ambulància. Imagineu que vosaltres sou els responsables de la logística per traslladar els ferits als hospitals. Per a no col·lapsar urgències, voleu seleccionar els hospitals de manera que cada hospital rebi com a màxim $\lceil n/k \rceil$ ferits.

    Dissenyeu un algorisme polinòmic per que un cop rebeu la informació sobre el lloc on es cada ferit, pugueu determinar si es possible que tots els ferits siguin transportats a un hospital de manera que arriben abans de 30 minuts i que cap hospital rebi més de $\lceil n/k \rceil$ ferits.

68. Tenim un digraf $G = (V, E)$ on cada $e \in E$ té capacitat $c(e) = 1$, i amb una font $s \in V$ i un sumider $t \in V$. També en donen un paràmetre $k \in \mathbb{N}$. Doneu un algorisme amb temps polinòmic per a resoldre el següent problema: Volem eliminar k arestes de G de manera que reduïm al màxim que puguem el flux $s \to t$. En altres paraules, volem trobar un $F \subseteq E$ tal que $|F| = k$ i el flux màxim $s \to t$ en $G' = (V, E - F)$ sigui el més petit possible.

69. Una gran part del potencial d'empreses com Yahoo!, Google, Amazon, etc. es basa en el fet que milions de persones visiten cada dia les seves pàgines, (en angles aquest fet s'anomena "eyeballs"). Un cop una persona visita una pagina web d' aquestes companyes, de manera subliminal (o no tant) se intenta convèncer per a que deixen informació personal, la qual cosa permet que en el futur, si la mateixa persona torna a visitar la mateixa pagina web, rebi informació i anuncis extremadament personalitzats i adreçats a l'usuari. Per exemple, si l'usuari ha transmès informació a Yahoo!, que té 20 anys i estudia a la UPC, a Barcelona, rebrà tota mena d'informació sobre lloguers d'habitacions a Barcelona, acadèmies etc. D'altra banda, si l'usuari és un alt executiu rebrà anuncis sobre cotxes de luxe, creuers per illes exòtiques o vacances a la lluna. Una tasca algorismica interessant és personalitzar els anuncis seleccionant els més adients a cada persona.
Suposem que els administradors d'una web popular han identificat k grups de característiques diferents (perfils) $G_1, G_2, \ldots, G_k$, que no són necessàriament disjunts, per exemple G_i pot ser viure a l'Hospitalet, G_j ser dona i G_k que estudia a la UPC. La companyia propietària de la web té contractes amb m empreses anunciadores per mostrar un nombre determinat de còpies dels seus anuncis als usuaris de les seves pagines web. El contracte que la companyia i-èsima fa amb Yahoo! és del tipus:

- Per a un subconjunt $X_i \subseteq \{G_1, \ldots, G_k\}$ dels grups demogràfics, i vol que els seus anuncis apareguin únicament a grups que són a X_i.
- Donat un enter r_i l'anunciant vol que els seus anuncis es mostren al menys a r_i usuaris.

Suposem que en un moment donat, hi ha n usuaris visitant la web. Com que tenim la informació de cadascun d'aquests usuaris, és fàcil conèixer el perfil de l'usuari j (per $j = 1, 2, \ldots, n$), es dir a quin subconjunt $U_j \subseteq \{G_1, \ldots, G_k\}$, pertanyé Voldríem dissenyar un mecanisme tal que cada usuari vegi un únic anunci (d'uns pocs segons) de manera que per se satisfan restriccions imposades per els m anunciants? És dir, per a cada $i = 1, 2, \ldots m$ com a mínim r_i usuaris, on cadascun pertanyé a almenys a un grup a X_i, veu un anunci proporcionat per l'anunciant i. Donar un algoritme eficient (polinòmic) per decidir si això és possible, i si és així, per a triar un anunci que aparegui per a cada usuari.

70. Considerem un model molt simplificat d'una xarxa de telefonia mòbil a zones rurals amb baixa densitat de població. Se'ns dona la ubicació de les n estacions base (antenes per a mòbil), especificades com a punts $b_1, \ldots, b_n$ al pla. També se'ns dóna la ubicació dels n telèfons mòbils, especificats com a punts $p_1, \ldots, p_n$ al pla. Finalment, se'ns dóna un paràmetre $\Delta > 0$ de distancia de cobertura dels mòbils Direm que el conjunt dels telèfons mòbils esta *completament connectat* si és possible assignar a cada telèfon a una estació base de manera que:

   - Cada telèfon s'assigna a una estació base diferent,
   - Si el telèfon al punt p_i s'assigna a una estació base b_j, llavors la distància en línia recta entre p_i i b_j és $\leq \Delta$.

   Suposem que el propietari del mòbil p_1 decideix fer un viatge en cotxe cap a l'est sense aturant-se, recorrent un total d'z unitats de distància . Com aquest telèfon mòbil es mou, s'ha d'anar actualitzant l'assignació del mòbil a diferents estacions base per tal de mantenir el mòbil connectat a la xarxa telefònica

   Doneu un algoritme polinòmic per decidir si és possible mantenir la connectivitat entre tot conjunt de mòbils en tot moment totalment connectats, durant el trajecte d'aquest telèfon. Podeu assumir que tots els altres telèfons romanen estacionaris durant aquest viatge. Si és possible mantenir la connectivitat, l'algorisme ha de produir una seqüència de l'assignació dels mòbils a les estacions base, que sigui suficient per a mantenir la connectivitat. Altrament, l'algorisme ha de tornar en quin punt (coordenada) del pla es perd la connectivitat. L'algorisme hauria de tenir una complexitat de $O(n^3)$.

   Exemple: Suposem que tenim 2 mòbils a $p_1 = (0, 0)$ i $p_2 = (2, 1)$; i tenim 2 estacions-base a $b_1 = (1, 1)$ i a $b_2 = (3, 1)$ amb $\Delta = 2$. Suposem que p_1 es desplaça 4 unitats cap l'est fins al punt $(4, 0)$ aleshores podem mantenir la connectivitat entre els mòbils, al començament assignem $p_1 \to b_1$ i $p_2 \to b_2$ i quan p_1 arriba a $(2, 0)$ n'assignem $p_1 \to b_2$ i $p_2 \to b_1$.

71. Sigui $G = (V, E)$ un digraf, i suposem que cada $v \in V$ té el mateix nombre d'arestes que entren que d'arestes que surten, o sigui $|\{(u, v) : (u, v) \in E\}| = |\{(v, u) : (v, u) \in E\}|$. Donat un enter $k \geq 1$, siguin $x, y \in V$ tal que existeixen k camins disjunts (sense repetir arestes) $x \to y$. Es cert que també existeixen k camins disjunts $y \to x$? Si la resposta és afirmativa, doneu una demostració, altrament doneu un contraexemple.

72. En sociologia, sovint s'estudia un graf $G = (V, E)$ en el qual els nodes representen les persones, i les arestes representen els que són amics entre si. Suposem que l'amistat és simètrica, per la qual cosa podem considerar G com un graf no dirigit.
Volem estudiar aquest graf G, per a trobar grups de persones, molt unides en el sentit de que tots són amics de tots. Una manera de formalitzar aquesta idea seria la següent. Per a un subconjunt $S \subseteq V$ sigui $e(S)$ el nombre d'arestes dintre d'S, és a dir, el nombre d'arestes que tenen tots dos extrems en S. Definim la *cohesió* d'S com $e(S)/|S|$. en aquest tipus de grafs, un paràmetre natural podia ser el conjunt $S \subseteq V$ amb cohesió màxima.

   (a) Doneu un algorisme polinòmic que pren com a entrada G i un nombre α racional, i determina si hi ha un conjunt S amb cohesió $\geq \alpha$.

   (b) Doneu un algoritme polinòmic per trobar un conjunt $S \subseteq V$ amb la màxima cohesió.

73. El *problema de la evacuación* tiene la siguiente definición. Nos dan un grafo dirigido $G = (V, E)$ que representa una red de carreteras. Una cierta colección de nodos $X \subseteq V$ se designan como *nodos habitados*. Otra colección de nodos S son designados como *nodos seguros*. Suponemos que S y X son disjuntos. En el caso de una emergencia, queremos conocer rutas de evacuación desde los nodos habitados a los seguros. Un conjunto de *rutas de evacuación* se define como un conjunto de caminos en G tales que: (i) cada nodo en X aparece como el inicio de un camino, (ii) el último nodo en un camino aparece en S, (iii) los caminos no comparten aristas.

    Cada ruta de evacuación proporciona una ruta a travéss de la cual los habitantes de un nodo poblado pueden escapar a un nodo seguro sin compartir ninguna parte del camino con habitantes de otros nodos.

    Un conjunto de *rutas de evacuacin mixtas* se define como un conjunto de caminos en el que (i) cada nodo en X aparece como el inicio de un camino, (ii) el último nodo en un camino aparece en S.

    Este tipo de rutas proporciona rutas de escape en las que parte del camino es compartido entre habitantes de distintas ciudades.

    Supongamos que además, debido a las restricciones de tráfico, cada arco $e \in E$ tiene asignada una capacidad de tráfico $c(e)$ que corresponde al número máximo de vehículos que pueden atravesarlo. Una petición de tráfico $r(x)$, para un nodo habitado x, representa el número de vehículos que se quieren evacuar desde x.

    (a) Dados $G = (V, E, c)$ y $S, X \subseteq V$, muestra como decidir en tiempo polinómico si un conjunto de rutas de evacuación existe o no. En caso de que si que existan, el algoritmo debe proporcionar, para cada nodo poblado, la capacidad del arco de capacidad mínima de la ruta que se inicia en él.

    (b) Dados $G = (V, E, c)$, $S, X \subseteq V$, y una petición de tráfico $r(x)$, para cada $x \in X$, muestra como decidir en tiempo polinómico si existe un conjunto de rutas mixtas de evacuación que permitan enviar los vehículos solicitados desde X a nodos seguros y que, además, para cada arco, se cumpla la condición de que el número de vehículos que lo atraviesan no supere la capacidad de tráfico del arco. En caso de que sea posible la evacuación con rutas mixtas, el algoritmo tiene que devolver además, para cada nodo $x \in X$, un plan de evacuación indicando: para cada vehículo sliendo de x, un camino con origen en x y final en un nodo seguro. El total de tráfico asignado entre todos los caminos del plan de escape de todos los nodos $x \in X$ tiene que cubrir la totalidad de las peticiónes de tráfico y cumplir las restricciones de capacidad de los arcos.

74. **SafeTrans** es dedica al transport de mercaderies perilloses o de gran volum i/o pes per carretera. La companyia està analitzant la possibilitat d'acceptar un encarreg per traslladar els residus emmagatzemats a un dipòsit cap a un altre més segur. Per fer l'anàlisi de viabilitat el seu equip logístic ha definit un graf dirigit $G = (V, E)$ que representa la xarxa de carreteres que permetria connectar els dos dipòsits $s, t \in E$ dintre d'uns límits de quilometratge raonables.

Degut a la natura dels materials a transportar el trasllat s'ha de fer al llarg de com a màxim 5 dies durant la nit i a més s'ha de tenir en compte, en la mesura del possible, el nivell de seguretat que es pot assolir. Per això volen que l'equip informàtic els doni informació sobre la possibilitat de fer el trasllat tenint en compte diferents restriccions. En tots els escenaris, per raons de seguretat un tram de carretera només es pot fer servir com a molt una de les 5 nits.

Escenari 1: Si un tram de carretera es fa servir una nit, aquesta nit només hi pot circular per ell un camió carregat.

Escenari 2: Les condicions de l'escenari 1 es poden relaxar, per un subconjunt de trams de carretera suficientment aïllats. Suposant que $S \subset E$ son els trams suficientment aïllats. En aquest escenari s'ha de garantir: Si un tram de carretera a S es fa servir una nit, aquesta nit poden circular un o dos camions carregats. Si un tram de carretera no és a S i es fa servir una nit, aquesta nit només hi pot circular un camió carregat.

Escenari 2: Una anàlisi alternativa de la densitat dels nuclis de població ha determinat que en la majoria dels casos la població està suficientment aïllada, però que hi han un cert subconjunt de nuclis urbans densament poblats. Suposant que $D \subset V$ és el conjunt de nuclis urbans amb alta densitat de població. En aquest escenari s'ha de garantir que com a molt un camió carregat circuli una nit per $d \in D$. En contrapartida, si un tram de carretera es fa servir una nit, aquesta nit poden circular per ell fins a tres camions carregats.

Proporcioneu algorismes, per a cada escenari, què:

(a) Donats, G, S i D juntament amb s i t, determini un conjunt tan gran com sigui possible de rutes potencials per fer el transport en una nit.

(b) Donats, G, S, D, c, $1 \leq c \leq 5$, i k juntament amb s i t determini si es possible seleccionar rutes per k camions i c dies, i en cas de que sigui possible proporcioni aquestes rutes, assegurant què: cada camió carregat fa com a molt un trajecte per nit; globalment es compleixen les restriccions imposades per l'escenari; i a més el nombre total de trasllats es més gran que $\frac{2}{3}ck$.

Solució:

(a) Resolveremos el problema como un problema de flujo máximo. Para cada escenario tendremos una red de flujo que modela el problema. Sobre esta red calcularemos el flujo máximo usando Ford Fulkerson. Una vez calculado un flujo con valor máximo consideraremos el grafo G' en el que solo aparecen las aristas con flujo positivo. En G' repetiremos el siguiente proceso: obtener un camino de s a t, este camino será una ruta; reducir el valor del flujo en una unidad en todas las aristas del camino; eliminar aquellas en las que el flujo sea 0.

    En el escenario 1, tenemos que calcular caminos que no comparten aristas, tal y como hemos visto en clase basta con asignar capacidad 1 a todos los arcos en E y tomat s como fuente y t como sumidero.

    En el escenario 2, los arcos en S pueden soportar dos caminos, les asignaremos capacidad 2 y a los que no están en S capaciadad 1. Cualquier conjunto de rutas que cumpla las condiciones se puede convertir en un flujo en el que el valor del flujo en arcos de S es menor o igual que 2 y al revés siempre que el flujo tenga valores enteros. Por lo tanto un flujo entero con valor máximo nos proporciona una solución al problema en este escenario.

    En el escenario 3, asignaremos a los arcos capaciadad 3 y tal como hemos visto en clase para conseguir que los caminos no pasen dos veces por el mismo node en D, convertiremos el grafo en un grafo G' donde cada nodo $d \in D$ se remplaza por dos nodos d' y d'', los arcos que entran en d entraran en d', los que salen lo harán de d'' y añadimos un arco (d', d'') con capacidad 1.

    En cualquiera de los tres casos el coste de calcular maxflow es $O((n+m)m)$ ya que el valor del flujo máximo es como mucho $3m$. Con el mismo coste se pueden obtener la lista de rutas, realizando un BFS por cada unidad de flujo.

(b) Notemos que los tramos de carretera solo se pueden utilizar en uno de los c días, por ello es necesario que el número total de rutas, calculado en el apartado a) sea $\geq \frac{2}{3}ck$.

    Además necesitamos encontrar una partición de E en c conjuntos $E_1, \ldots, E_c$ que nos permita asignar $\leq k$ rutas por día y suficientes rutas a lo largo de los c días. Para una partición $E_1, \ldots, E_c$, resolveremos el problema de flujo máximo como en el apartado a) obteniendo c valores de flujo $f_1, \ldots, f_c$, dónde $f_i = maxflow(V, E_i, s, t, c)$. La partición nos proporciona una solución sí $\sum_{i=1}^{c} \min k, f_i \geq \frac{2}{3}ck$.

    En el escenario 1 las rutas son totalmente independientes, ya que no comparten aristas y por lo tanto se pueden asignar a cualquiera de los dias. Una vez resuelto a) asignamos las primeras k rutas al día 1, las siiguientes k al día 2, hasta que acabemos con todas las rutas disponibles o llenemos los c días.

    En los escenarios 2 y 3, recorreriamos todas las particiones posibles hasta encontrar una en la que las condiciones se cumplan o concluir que no hay tal solución.

En el escenario 1 el coste es como en el caso a) $O((n+m)m)$, pero podríamos reducirlo a $O((n+m)ck)$ finalizando el algoritmo cuando el valor del flujo llegue al mínimo requerido. En los escenarios 2 y 3 tenemos que recorrer todas las particiones, este número es exponencial si $c > 1$, el algoritmo tiene coste exponencial, aún cuando el coste por partición es polinómico.

75. El servicio de control alimentario de la Generalitat está intentando acreditar un procedimiento de medida de la calidad alimentaria de productos complejos. En el laboratorio disponen m cromatógrafos y tienen técnicas de análisis de n componentes. Para cada cromatógrafo j se conoce conjunto S_i de componentes que puede analizar.

    Para acreditar el procedimiento necesitan garantizar que se han realizado como mínimo k análisis en cromatógrafos diferentes de cada una de las componentes. Dependiendo del nivel de independencia se requieren condiciones adicionales.

    - Nivel A: Un cromatógrafo solo puede utilizarse para analizar un máximo de dos componentes.
    - Nivel B: Los cromatógrafos del laboratorio son de tres marcas diferentes. En este nivel se requiere además que los cromatógrafos que analicen una componente no sean todos de la misma marca.

    Diseñad algoritmos para resolver los siguientes problemas:

    - Dados n, m, k y para cada cromatógrafo j, $1 \leq j \leq m$, los conjuntos $S_j \subseteq \{1, \ldots n\}$, determinar si es posible realizar análisis de las n componentes con el nivel de acreditación A.
    - Dados n, m, k y para cada cromatógrafo j, $1 \leq j \leq m$, los conjuntos $S_j \subseteq \{1, \ldots n\}$ y la marca $m_j \in \{1, 2, 3\}$, determinar si es posible realizar análisis de las componentes con el nivel de acreditación B.

    Los dos algoritmos, siempre que sea posible, deben obtener además una asignación de cromatógrafos a componentes verificando las condiciones requeridas por el nivel de acreditación.

76. Un grup d'amics lloguen entre tots una embarcació d'esbarjo per sortir junts a navegar. Cada vegada que es fa un viatge cal que un dels usuaris (el mateix durant tot el viatge) s'encarregui de preparar el vaixell, n'assumeixi la conducció i l'atracada. També haurà de fer-se responsable de la neteja i reparació de desperfectes apareguts durant el viatge. A aquesta persona l'anomenem *responsable* del viatge. A ningú li ve de gust ser responsable i la nostra tasca és fer una assignació *justa* respecte de l'ús que fa cadascuna de les persones del grup.

   Hi han dos elements que semblen raonables a l'hora de definir una assignació justa. Si una persona fa servir el vaixell moltes vegades, també hauria de ser responsable moltes vegades. D'altra banda, si una persona fa servir el vaixell en dies en què poques persones volen usar-lo, això també hauria d'incrementar el nombre de vegades que és responsable. Si no veus clar aquest segon criteri, pensa en el cas que 2 persones viatgen 20 vegades juntes i amb ningú més. És lògic que en siguin responsables 10 vegades cadascuna. Si per altra banda hi ha 20 persones que viatgen 20 vegades juntes, i amb ningú més, és natural que els toqui ser responsable un cop a cadascuna d'elles.

   Suposem que una persona viatja r vegades. En el seu primer viatge hi ha k_1 passatgers, en el segon k_2, i així fins k_r. Sembla just que a aquesta persona sigui responsable $S = \sum_{j=1}^{r} \frac{1}{k_j}$ vegades. Com que aquest número pot no ser sencer, ens conformarem amb que sigui responsable com a molt $\lceil S \rceil$ vegades. Si això passa per a tothom, direm que la assignació és justa.

   Assumim que el grup està format per un conjunt A de m amics, $A = \{1, \dots m\}$. L'entrada del problema correspon a la informació de n viatges. Per a cada viatge i, s'ens dona el conjunt $V_i \subseteq A$ de viatgers.

   (a) Demostreu que per a qualsevol entrada $V_1, \dots, V_n$, hi ha una assignació justa.

   (b) Proporcioneu un algorisme de cost polinòmic en n i m que proporcioni una assignació de responsable per a cadascun dels n viatges que sigui justa.

77. Es vol emmagatzemar en un dispositiu de capacitat L un conjunt de n arxius. La talla del arxiu i-èsim ve donada per ℓ_i. Se sap que $\sum_{i=1}^{n} \ell_i > L$. Dissenyeu un algorisme que seleccioni un conjunt d'arxius tot maximitzant el nombre d'arxius que es poden emmagatzemar al dispositiu. Justifiqueu la correctesa del vostre algorisme i analitzeu el seu cost.

78. Considereu un examen amb n qüestions, on la qüestió i-èsima té un valor de $v_i > 0$ punts i es necessiten m_i minuts per a resoldre-la. Necessitem un total de V punts per aprovar. Dissenyeu un algorisme que donats $v_1, v_2, \ldots, v_n$ i $m_1, m_2, \ldots m_n$ compute el mínim temps per a obtenir com a mínim V punts. En particular,

(a) Si $M(i, v)$ denota el mínim nombre de minuts necessaris per obtenir v punts, quan únicament teniu qüestions $1, \cdots, j$, doneu la recurrència per a $M(i, v)$.

(b) Dissenyeu un algorisme per a resoldre el problema en temps $O(nV)$. Quina n'és la complexitat? Digueu com a partir del vostre algorisme previ, podeu escriure la llista $S \subseteq \{1, 2, \ldots, n\}$ de qüestions a resoldre tal que $V \leq \sum_{i \in S} v_i$ i $\sum_{i \in S} v_i$ es minimitza.

(c) Suposem que pressionat per l'autoritat acadèmica, el professor decideix incrementar els aprovats donant punts parcials a cada pregunta, on la puntuació és proporcional als minuts que l'alumne ha escrit, de la següent manera, a la pregunta i-èsima per cada minut d'escriptura, l'alumne obté v_j/m_j punts (fins a un màxim de v_i punts). Dissenyeu un algorisme per a determinar en quines qüestions heu d treballar i quan de temps heu de treballar a cadascuna, per a obtenir més de V punts. Demostreu la correctesa i doneu la complexitat.

79. La firma *Doctors on Call* té que resoldre el següent problema. Per a cadascun dels pròxims n dies, la firma ha determinat el nombre de doctors disponibles que requereix. Així al dia i-èsim, necessiten exactament p_i doctors. Hi han k doctors en total, i cadascú d'ells ha donat una llista amb els dies en que està disposat a treballar. Així el doctor j proporciona un conjunt L_j de dies. Doctors on Call vol, a partir d'aquesta informació, un procediment que permeti tornar a cada doctor j una llista definitiva de dies L'_j amb les propietats següents: (1) el conjunto $\Delta_j = L'_j \setminus L_j$ té com a molt c dies; i (2) quan es considera tot el conjunto de listas $L'_1, \ldots, L'_k$, per a cada dia $1 \leq i \leq n$, hi han exactament p_i doctors que tenen el dia i a la seva llista definitiva. El paràmetre c reflecteix la tolerància de l'assignació i pot variar segons las circumstancies. Per suposat, si tal solució no es possible, el sistema ha de (correctament) informar de que aquest és el cas.

   (a) Proporcioneu un algorisme de cost polinòmic en n i k que resolgui el problema per un valor de tolerància c donat.

   (b) Proporcioneu un algorisme de cost polinòmic que obtingui la tolerància mínima per que el problema tingui solució, assumint que $p_i \leq k$ per $1 \leq i \leq n$.

80. Considereu el joc de cartes següent : Es posen una seqüència de cartes, $a_1, \ldots, a_n$ cara amunt (on n és parell). La carta a_i té un valor $v_i \in \mathbb{N}$. La seqüència està desordenada El joc consisteix a que cada jugador escull per torns una de les dues cartes possibles; la primera o la darrera de la seqüència. Guanya el jugador que aconsegueix que el valor de les seves cartes sigui el més gran. Si únicament hi juguen 2 jugadors:

(a) Doneu una seqüència on l'estratègia del que juga primer no és escollir la carta amb valor més alt.

(b) Doneu un algorisme amb el millor temps que pugueu, que computi una estratègia òptima per a la persona que juga primer (estratègia que li faci guanyar sempre) (Ajut: Definiu un sub-joc, com el joc que ens queda per jugar, quan s'han escollit i cartes per l'esquerra i j cartes per la dreta $(i + j \leq n)$).

(c) Quina és la complexitat del vostre algorisme?

81. SuperFast és una empresa de transport que està intentant decidir si li interessa participar en una oferta de treball de Amazon a Barcelona. Amazon voldria subcontractar el transport de productes de proximitat a SuperFast. El transport ha de garantir uns compromisos de puntualitat molt estrictes i SuperFast vol una estimació de la quantitat de nous vehicles que hauria d'incorporar a la seva flota i una estimació del cost corresponent al seu ús. Amazon li ha proporcionat els resultats de les seves simulacions de comportament dels clients i, en particular, de les necessitats de transport diàries per uns quants dies. Per un dia es disposa d'una llista de sol·licituds de transport. Cada sol·licitud de transport especifica les coordenades GPS de l'origen i del destí d'un enviament juntament amb l'hora de recollida i d'entrega. SuperFast disposa d'un programa que li proporciona el temps necessari de desplaçament d'un vehicle entre qualsevol parell de posicions de la ciutat coneixent el temps d'inici del trasllat.

    En aquest primer estudi SuperFast assumeix el cas pitjor en el què els vehicles no poden portar més d'un enviament. A més, el vehicle ha de ser a la posició d'origen al temps estipulat i no pot deixar el punt de destí fins el temps estipulat de trasllat. Així, un vehicle que transporti un enviament pot encarregar-se d'un altre sempre que el temps de desplaçament entre el destí i el nou origen li permeti arribar l'hora estipulada. També assumeix que l'estimació del temps necessari de desplaçament és acurada.

    Disenyeu algorismes amb cost polinòmic que:

    (a) Donats un nombre k i les sol·licituds de transport d'un dia, determini si es poden servir amb k vehicles.
    (b) Donades les sol·licituds de transports d'un dia, determini el nombre mínim de vehicles que les poden servir ($k_{\min}$)
    (c) Donades les sol·licituds de transports de un dia, proporcioni una planificació per a cadascun dels $k_{\min}$ vehicles indicant, per a cada vehicle, la seqüència de sol·licituds de transport que ha de servir i el temps total de desplaçament del vehicle.

    Podeu suposar que el càlcul del temps de desplaçament entre dos posicions té un temps constant i que entre l'hora de recollida i la d'entrega hi ha temps suficient per fer-hi el desplaçament amb puntualitat.

    Ajut: Podeu pensar un vehicle com una unitat de flux i una sol·licitud de transport com un arc amb fites inferiors i superiors a la seva capacitat.

    Una solució

    (a) Utilizando la ayuda identificaremos los vehiculos con unidades de flujo y las solicitudes con arcos con capacidad limitada con cota superior e inferior.

        De las restricciones del problema sabemos que un vehículo puede servir otro envío siempre que pueda desplazarse con tiempo suficiente desde el destino al nuevo origen, asumiendo que permanece en el destino hasta la hora estipulada. Esta restricción nos indica cual es la red a considerar.

Tendremos, para cada solicitud i, dos nodos o_i y d_i y un arco (o_i, d_i) con capacidad $[1, 1]$, reflejando el hecho de que todas las solicitudes tienen que ser servidas. Añadiremos un arco (d_i, o_j) cuando saliendo del destino de la solicitud i en el tiempo estipulado podamos llegar al origen de la solicitud j en el tiempo estipulado, reflejando la observación previa.

Añadiremos tres vertices adicionales s, x, t y los arcos (s, x) con capacidad k, para forzar que el flujo máximo sea $\leq k$, i, para cada solicitud i, los arcos (x, o_i) i (d_i, t) con capacidad 1.

Notemos que la existencia de un flujo que satisface las restricciones de capacidad nos garantiza la existencia de $\leq k$ caminos que no comparten aristas de s a t que, además, cubren todos los arcos correspondientes a solicitudes. Esto es equivalente a poder servir las peticiones con k o menos vehículos.

Una vez obtenida la red de flujo, podemos determinar si hay un flujo que satisface las restricciones en tiempo polinómico utilizando el algoritmo visto en clase para decidir la existencia de flujos con demandas y cotas inferiores.

(b) Siempre hay una solución con n vehículos, por ello $1 \leq k_{\min} \leq n$ podemos implementar una búsqueda binaria utilizando el algoritmo del apartado (a) que nos permite decidir si las solicitudes se pueden servir con $\leq k$. En total tendermos que hacer $O(\log n)$ ejecuciones del algoritmo del apartado (a).

(c) Una vez encontrado el valor de $k_{\min}$ utilizando el algoritmo del apartado (a) para este valor de k tendremos una flujo de s a t con valor del flujo $k_{\min}$ que satisface la restricción de que todas las solicitudes están servidas. Como hemos visto en clase lo único que tenemos que hacer es aplicar el algoritmo de extracción de caminos disjuntos como vimos en clae. Esto nos da $k_{\min}$ ejecuciones de BFS sobre el grafo formado por los arcos con flujo positivo.