61. Sigui $G = (V, E)$ un graf dirigit amb pesos $w : V \to \mathbb{Z}^+$, donats dos vèrtexs $u_1, u_2 \in V$ definim el pes d'un camí $w(P(u_1, u_2))$ com $\sum_{v \in P(u_1,u_2)} w(v)$. Si tenim com a entrada $G, w, u_1, u_2$, doneu un algorisme per a calcular el camí amb menys pes entre $u_1$ i $u_2$. Quina és la seva complexitat? (Ajut: transformeu $G$ en $G'$ on els pesos siguin a les arestes)

    **Una solució** Farem servir l'ajut: construir un nou graf $G'$ idèntic al $G$, excepte que que per a tota aresta $(\vec{u, v})$ definim $w(\vec{u, v}) = w(v)$. El cost de crear $G'$ és $O(n+m)$. Com els pesos són positius, podem utilitzar Dijkstra per calcular el camí més curt entre $u_1$ i $u_2$ amb un cost $O(m \lg n)$ (utilitzant un heap) o podem utilitzar Bellman-Ford amb un cost $O(nm)$. Per a demostrar que el camí més curt a $G'$ també és el camí més curt a $G$, sigui $u_1, v_1, \ldots, v_k, u_2$ un camí amb pes $w(u_1) + w(v_1) + \cdots + w(v_k) + w(u_2)$ a $G$, el mateix camí a$G'$ tindrà pes $w(\vec{u_1, v_1}) + \cdots + w(\vec{v_k, u_2})$ que és $= w(v_1) + \cdots + w(v_k) + w(u_2) \Rightarrow$ els pesos de qualsevol camí en $G$ i $G'$ es diferencien en $w(u_1)$, per tant un camí amb distancia mínima a $G'$ també serà un camí amb distancia mínima a $G$.

62. Donada una cadena $x \in \{0,1\}^n$, escrivim $x^k$ per a representar $x$ copies de $x$ concatenades (una darrera l'altra) Direm que una cadena $x'$ és una repetició de $x$ si existeix un $k \in \mathbb{N}$ tal que $x'$ és un prefix de $x^k$ (per ex. $x' = 10110110110$ és una repetició de $x = 101$.

Diem que una cadena $s$ és una *trena* de $x$ i $y$ si podem particionar els símbols de $s$ en dues subseqüències $s'$ i $s''$, no necessàriament contigues, de manera que $s'$ és una repetició de $x$ i $s''$ és una repetició de $y$. Es a dir, cada símbol de $s$ ha de ser a $s'$ o a $s$". Per exemple, si $x = 101$, $y = 00$, i $s = 100010101$. $s$ és una trena de $x$ i $y$, ja que els símbols a les posicions 1,2,5,7,8,9 (=101101) són un repetició de $x$, i la resta dels símbols forment 000 que és una repetició de $y$).

Doneu un algorisme eficient tal que donats $x, y$ i $s$ decideixi si $s$ és una trena de $x$ i $y$.

**Una Solució:**

Primer de tot establirem notació i definicions auxiliars.

Si $w = w_1 \cdots w_n$, $w[k]$, per $1 \leq k \leq n$, és la subcadena formada per els primers $k$ caràcters de $s$.

$t[w,i] = w^i[i]$ és el prefixe de longitud $i$ de $w^i$

Utilizarem $V$ per a representar "cert".

Sigui $x = x_1x_2 \ldots x_p$, $y = y_1y_2 \ldots y_q$ i $s = s_1s_2 \ldots s_n$ una entrada del problema a resoldre.

Per a establir una recurrència que ens permeti resoldre'l considerarem un problema auxiliar que resoldrem recursivament. Aux: determinar si la subcadena $s[k]$, $1 \leq k \leq n$, és una trena de $t[x,i]$ i $t[y,j]$ per a tots els valors possibles de $i, j, k$ on $k = i + j$.

Volem calcular una taula $C(i,j)$, $i+j \leq n$, tal que $C(i,j) = V$ sii $s[i+j]$ és una trena de $t[x,i]$ y $t[y,j]$ .

Aquesta condició és equivalent a dir que $s[i+j]$ es pot dividir en $s'$ ($|s'| = i$) una repetició de $x$ i $s''$ ($|s''| = j$) una repetició de $y$.

Observem que quan la descomposició és possible l'últim caràcter de $s[i+j]$ ha de coincidir amb l'últim caràcter de $t[x,i]$ o de $t[y,j]$ (o amb els dos). En cas contrari la descomposició no és possible. Al primer cas, $s_{i+j} = x_{i'}$, per $i' = i \mod p$ i tenim, a més, què $[s[i+j-1]$ ha de ser una trena de $t[x,i-1]$ i $t[y,j]$. Al segon cas, $s_{i+j} = y_{j'}$, per $j' = j \mod q$ i tenim què $[s[i+j-1]$ ha de ser una trena de $t[x,i]$ i $t[y,j-1]$.

Aquesta observació sobre suboptimalitat de les solucions ens porta a la recurrència:

$$C(i,j) = [(s_{i+j} = x_{i \mod p}) \vee C(i-1,j)] \wedge [(s_{i+j} = y_{j \mod q}) \vee C(i,j-1)],$$

amb $C(0,0) = V$; per $j \in [n]$, $C(0,j) = V$ sii $s_1s_2 \ldots s_j$ és una repetició de $y$; per $i \in [n]$, $C(i,0) = V$ sii $s_1s_2 \ldots s_i$ és una repetició de $x$.

La resposta final de l'algorisme és $\vee_{i+j=n, i \mod p=0, j \mod q=0} C(i,j)$.

El temps total per implementar aquest algorisme recursiu amb un esquema de PD és $O(n^2)$ ja que el cost per element és constant.

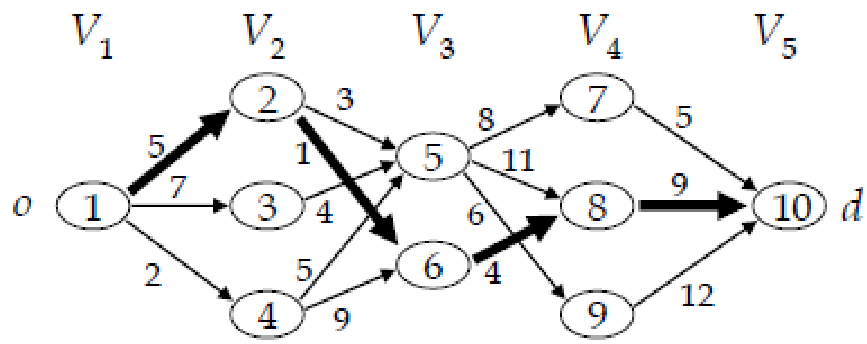
Per finalitzar s'ha de fer un recorregut de la diagonal amb suma $n$ de la matriu $C$ acumulant els valors de les posicions que ens interessin. Això ens dona un temps addicional de $O(n)$.

Per tant el cost total és $O(n^2)$ i fem servir espai $O(n^2)$.

63. Els estudiants de la FIB volen dissenyar una xarxa social (i.e. un graf dirigit $G = (V, E)$) per determinar el grau de simpatia entre tota la comunitat universitària a la UE. El graf es dissenya a partir de relacions personals; si $a$ coneix $b$, $a, b \in V$ i $(a, b) \in E$. A més, a cada aresta $(a, b)$ se li assigna un pes entre 0 i 10 que indica la simpatia de $b$ en opinió de $a$ (0 molta antipatia, 10 molta simpatia).

    Per tal que un estudiant $a$ pugui tenir una idea del grau de simpatia d'un estudiant $d$ que no coneix, simplement ha de trobar el valor del camí amb pes màxim $\mu(a, d)$ i el valor del camí amb pes mínim $\delta(a, b)$. Però hi ha un problema no sabem com trobar el valor del camí amb pes màxim. Per sort hi ha un estudiant de l'assignatura d'Algorísmia de la FIB té una l'idea: negar el valor dels pesos (i.e. si una aresta té pes 7, assignar-li el pes -7) i aplicar Bellman-Ford per a trobar el camí mínim, que serà el màxim sense negar. Penseu que l'algorisme del vostre col·lega és una bona solució?

64. Camí més curt en DAG multi-etapa. Un graf dirigit acíclic (DAG) $G = (V, E)$ és multi-etapa quan els vèrtexs es poden dividir en $k$ subconjunts $V = V_1 \cup V_2 \cup \cdots \cup V_k$ tal que cada arc va des un subconjunt $V_i$ al següent$V_{i+1}$. A més, el primer i últim subconjunt contenen exactament un vèrtex cadascun (és a dir, $V_1 = \{o\}$ i $V_k = \{d\}$ als que direm font i embornal, respectivament. A la figura següent teniu un exemple de DAG multi-etapa ponderat.




Dissenyeu un algorisme que calculi el camí més curt a partir de $o$ a $d$ en un DAG multi-etapa ponderat.

65. Considerem el problema d'imprimir de manera polida una frase amb una impressora. El text d'entrada és una seqüència de $n$ mots amb longitud $l_1, l_2, \ldots, l_n$, on cada longitud ve donada en caràcters. Cada línia pot contenir com a màxim $M$ caràcters. Realitzem la impressió de manera que si una línia conté els mots de $i$ fins a $j$, on $i \leq j$, i deixem exactament un espai entre mots, aleshores el nombre de caràcters en blanc al final de cada línia és $M - j + i - \sum_{k=i}^{j} l_k$, que ha de ser no-negatiu. Volem minimitzar la suma, sobre totes les línies excepte la darrera, dels quadrat d'aquestes magnituds. Dissenyeu un algorisme per imprimir de la manera indicada, un paràgraf amb $n$ mots. Analitzeu les complexitats espacials i temporals del vostre algorisme.

66. Considerem una xarxa d'agents mòbils, connectats per radiofreqüència, que tenen una distància curta de comunicació $r$. Aquestes xarxes es poden representar per un tipus especial de grafs anomenats *random geometric graphs* on els agents venen representats per nodes, i dos agents $x$ i $y$ estan connectats si $x$ cau dintre del cercle amb centre $y$ i radi $r$. Aquest tipus de xarxes són dinàmiques, per tant en l'instant $t$ pot haver-hi una aresta $(x, y)$, i en l'instant $t+1$, l'aresta pot deixar d'existir (cada agent pot sortit en direccions contràries). Volem dissenyar un sistema eficient, per mantenir un camí de connectivitat entre dos agents especificats. Volem que el camí sigui curt, però volem que aquest camí no canvii encara que el graf perdi i guanyi arestes. Sigui $V$ el conjunt de nodes que representen els agents, i en l'instant $t$ sigui $E_t$ el conjunt d'arestes. Per tant, als instants $t = 0, 1, \ldots, m$ les arestes evolucionen $E_0, E_1, \ldots E_m$, cosa que a cada $t$ ens dona un graf $G_t = (V_t, E_t)$. La xarxa d'agents té una representació temporal com una seqüència $\{G_i\}_t$ de grafs, sobre el mateix conjunt $V$ de nodes. Assumim que cada $G_i$ és connex.

Considerem dos nodes particulars $s, t \in V$. Sigui $P_i$ un camí mínim de $s$ a $t$ a $G_i$, i sigui $|P_i|$ la seva longitud (nombre d'arestes). Volem tenir una seqüència de camins $P_0, P_1, P_2, \ldots, P_m$ un camí mínim per a cada $G_i$. Definim el nombre de canvis $c(P_1, P_2, \ldots, P_m)$ a la seqüència com el nombre de índexs $i$, $(1 \leq i \leq m-1)$ per als quals $P_i \neq P_{i+1}$. L'objectiu es minimitzar aquest nombre de canvis, ja que a cada canvi tindrem un cost extra.

Sigui $k$ una constant donada, definim el *cost* de $P_0, P_1, P_2, \ldots, P_m$ com

$$CT(P_0, P_1, P_2, \ldots, P_m) = \sum_{i=1}^{m} |P_i| + k \cdot c(P_1, P_2, \ldots, P_m).$$

(a) Suposeu que existeix un únic camí $P$, que és el mateix camí entre $s$ i $t$ a $G_0, G_1, G_1, \ldots, G_m$. Doneu un algorisme polinòmic per a trobar-lo.

(b) Doneu un algorisme polinòmic per a trobar una seqüència de camins $P_0, \ldots, P_m$ amb cost mínim, on $P_i$ és un camí mínim de $s$ a $t$ a $G_i$.

67. A la universitat de Kakia, l'equip de govern està molt preocupat per l'efecte que els exàmens produeixen sobre l'estat anímic dels estudiants, per tant han decidit convertir l'edifici que alberga el Centre de Matemàtica utilitzant Neurones (CMN) en un centre d'esplai per als estudiants. EL Personal docent i investigador (PDI) allotjat a l'edifici del CMN, ha decidit defensar-se del desallotjament institucional i tancar-se a l'edifici. Per a aconseguir el desallotjament, l'equip de govern vol construir un eixam (swarm) de microrobots que ataquen al PDI a l'edifici fins que marxen. Els microrobots ataquen de la manera següent:

    (a) Durant $n$ segons, un eixam de robots arriba de manera que a l' $i$-èsim segon, arriben $x_i$ robots. EL PDI del CMN ha col.locat sensors envoltant l'edifici, de manera que poden preveure la seqüència $x_1, x_2, \ldots, x_n$ abans que els primers robots arribin.

    (b) El personal del CMN ha desenvolupat un polsador electromagnètic que pot destruir alguns dels robots quan arriben, el nombre de robots que destrueixen depèn del nivell de càrrega que tingui el polsador. Formalment, existeix una funció $f$ tal que si han transcorregut $j$ segons des de la darrera vegada que es va utilitzar el polsador, es destrueixen $f(j)$ robots. Per tant, si utilitzem el polsador al $k$-èsim segon, quan feia $j$ segons que s'havia utilitzat, el nombre de robots que destruiran serà $\min(x_k, f(j))$, i s'esgotarà la càrrega del polsador.

    (c) Al començament el polsador està totalment carregat, per tant si el polsador s'utilitza per primer cop al $j$-èsim segon, pot destruir $f(j)$ robots.

    Donada la informació $x_1, x_2, \ldots, x_n$ y donada la funció $f$, volem escollir els moments en què haurem d'utilitzar el polsador per a destruir el màxim nombre possible de robots. Per exemple, si $n = 4$, $x_1 = 1, x_2 = 10, x_3 = 10, x_4 = 1$ i $f(1) = 1, f(2) = 2, f(3) = 4, f(4) = 8$ aleshores la millor solució és activar el polsador al 3er i 4rt segon, al 3er segon destrueix 4 robots i al 4rt segon destrueix 1 robot (per la càrrega). En total es poden destruir 5 robots.

    Dissenyeu un algorisme eficient tal que donats $x_1, x_2, \ldots, x_n$ i $f$, retorni la seqüència de pulsacions que maximitzi el nombre de robots destruïts.

68. 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 ⇒ 741 Euros ⇒ 1,012.206 Canadian dollars ⇒ \$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.

69. 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.

70. Volem planificar una serie d'esdeveniments al llarg d'un període de temps. El sistema només permet expressar restriccions del tipus: L'esdeveniment $i$ es realitza almenys $x$ dies després de l'esdeveniment $j$, o bé del tipus: l'esdeveniment $i$ es realitza no més de $y$ dies abans de l'esdeveniment $k$. Donat un conjunt de restriccions d'aquests dos tipus, per a un conjunt de $n$ esdeveniments, doneu un algorisme que proporcioni una planificació que respecti les restriccions (si n'hi ha una). Analitzeu-ne el seu cost.

    Ajut: Penseu com podeu formalitzar les restriccions donades amb un graf amb pesos.

71. 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 després 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.

72. 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

73. 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.

74. 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.

75. Ara que a Catalunya les eleccions s'apropen, hi ha maneres d'agrupar els districtes electorals en circumscripcions electorals de manera molt acurada per tal d'arribar a resultats que afavoreixin a un partit polític en particular. Aquesta "tècnica"rep el nom de *gerrymandering* als USA i es pot traduïr com *districtació*.

    Gràcies al software disponible, la *districtació* ha canviat de ser una activitat portada a terme per un grup de gent amb mapes, llapis i paper a ser un procés automàtic. Actualment, la districtació no és més que un problema computacional (a més de ser molt susceptible al frau electoral). Es tenen bases de dades molt acurades per al seguiment demogràfic dels possibles votants, que arriben a poder conèixer el perfil de votant fins a nivell de carrer i fins i tot de vivendes. Aquestes dades es poden processar amb ordinadors per agrupar els votants en demarcacions "apropiades"als interessos del partit.

    Suposem que tenim un conjunt de $n$ districtes $D_1, D_2, \ldots, D_n$, cadascun amb $m$ votants. Se suposa que aquests $n$ districtes es reagrupen en dos circumscripcions, i que cadascuna aglutina $n/2$ districtes. Per a cada districte tenim informació sobre quants electors votaran a cada partit (per a simplificar, suposem que només tenim dos partits R i P). Es diu que $D_1, D_2, \ldots, D_n$ és *susceptible de frau electoral* si és possible realitzar la divisió en dos circumscripcions de tal manera que el mateix partit s'assegura la majoria en les dos circumscripcions.
    Doneu un algorisme per determinar si un determinat conjunt de districtes *és susceptible de frau electoral*. Doneu el temps d'execució del vostre algorisme, que ha de ser polinòmic en $n$ i $m$.
    **Exemple:** Suposem que tenim $m = 400$ votants, $n = 4$ districtes i la informació següent sobre els votants: sigui $P_i$ ($R_i$, respectivament) el nombre de votants del partit $P$ ($R$) al districte $i$. Si $P_1 = 55$, $R_1 = 45$, $P_2 = 43$, $R_2 = 57$, $P_3 = 60$, $R_3 = 40$, $P_4 = 47$, $R_4 = 53$, és possible repartir els districtes de manera que el partit $P$ tingui la majoria en les dues circumscripcions: simplement agrupem el districtes $D_1$ i $D_4$ que formen la circumscripció $C_1$ , i agrupem els districtes $D_2$ i $D_3$ en la circumscripció $C_2$. Per tant, aquest conjunt de districtes és susceptible de fraud. I malgrat que $P$ té en la població total una escassa majoria de 205 contra 195, acaba guanyant en les dues circumscripcions $C_1$ i $C_2$.
    Noteu, que si $P_1 = 90$, $R_1 = 10$, $P_2 = 45$, $R_2 = 55$, $P_3 = 45$, $R_3 = 55$, $P_4 = 45$, $R_4 = 55$, aquest conjunt de districtes no és susceptible de frau electoral.

76. 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 $p_2 \to b_1$.

77. 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.

78. 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 saliendo 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.

79. 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.

80. 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.

81. 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 $S_j \subseteq \{1, \dots n\}$, 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 $S_j \subseteq \{1, \dots n\}$ 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.

82. 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.

83. **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.