Algorísmia QP 2017–2018

Una possible solució al Segon Examen parcial **15 de Juny de 2018**
Durada: 2h

Instruccions generals:

- Els exercicis 1 i 2 s'han de resoldre fent servir l'espai reservat per a cada resposta.
- Heu d'argumentar la correctesa i l'eficiència dels algorismes que proposeu. Per això podeu donar una descripció d'alt nivell de l'algorisme amb les explicacions i aclariments oportuns que permetin concloure que l'algorisme és correcte i té el cost indicat.
- Heu de justificar totes les vostres afirmacions, en cas contrari la nota de la pregunta serà 0.
- Podeu fer crides a algorismes que s'han vist a classe, però si la solució és una variació n'haureu de donar els detalls.
- Es valorarà especialment la claredat i concisió de la presentació.
- Entregueu per separat les vostres solucions de cada bloc d'exercicis (Exer 1+2, Ex 3 i Ex 4).
- La puntuació total d'aquest examen és de **10 punts**.

Cognoms, Nom Data

Exercici 1 (2 punts) Digueu si les següents afirmacions són certes o no.

(a) (0.5 punts) El temps per tal de resoldre un problema de programació dinàmica sempre és $\Theta(k)$, on k és el nombre de subproblemes diferents del problema.

Sol. FALS. Hem vist molts problemes que requereixen cost no constant per subproblema.

(b) (0.5 punts) Donat un graf dirigit $G = (V, E)$ amb pesos sobre les arestes $w : E \to \mathbb{Z}^+$, l'algorisme de Dijkstra i l'algorisme de Bellman-Ford poden produir diferents arbres de camins mínims, malgrat que la mínima distancia entre dos vèrtexs qualsevol serà la mateixa.

Sol. CERT. Mirar les transparencies.

(c) (0.5 punts) Sigui $G = (V, E)$ un graf dirigit amb pesos sobre les arestes $w : E \to \mathbb{Z}^+$ i siguin $c, a \in V$. Aleshores el camí més curt de c a a no canvia quan incrementem en 1 el pes de totes els arcs.

Sol. FALS. Considereu un graf amb la aresta (c, a) amb pes 4 i un camí de longitud 3 de c a a amb totes les arestes amb pes 1 (total 3). A l'incrementar en 1, tenim el camí d'una aresta amb pes 5 i el de tres amb pes total 6.

(d) (0.5 punts) Si a una xarxa de flux $\mathcal{N}$ incrementeu la capacitat d'un arc en 1 unitat, sempre incrementarem el flux màxim en 1 unitat.

Solució: Fals, considereu la xarxa $s \to a \to t$ on cada aresta té capacitat=1. Incrementar la capacitat d'una de les dos arestes no incrementa el flux.

Exercici 2 (2.5 punts)

- (1.5 punts) 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?

Solució: FALS (no). Quan el graf esdevingui un xic dens, hi poden apareixer cicles amb pes negatiu .

- (1 punt) Doneu un algoritme amb cost $O(|V|\,|E|)$ per determinar si un graf dirigit $G = (V, E)$ amb pesos sobre les arestes $w : E \to \mathbb{Z}$ té un cicle amb pes negatiu.

Solució: Bellman Ford (BF) nos permite obtener decidir si G tiene un ciclo negativo accesible desde el nodo seleccionado con coste $O(|V|\,|E|)$. En principio tendríamos que ejecutar BF $|V|$ veces. Para evitarlo construimos grafo G' añadiendo a G un vértice nuevo s y un arco (s, u) para cada $u \in V$. Como en s no entra ningún arco s no puede formar parte de ningún ciclo. Además cualquier ciclo en G es accesible desde s. Ejecutando BFen G' desde s podemos resolver el problema en el tiempo requerido.

Exercici 3 (2.5 punts) Considereu el problema d'emmagatzemar n llibres als prestatges de la biblioteca. L'ordre dels llibres és fixat pel sistema de catalogació i, per tant, no es pot canviar. Els llibres han d'aparèixer a les prestatgeries en l'ordre designat. Les prestatgeries d'aquesta biblioteca tenen amplada L i són regulables en alçada. Un llibre b_i, on $1 \leq i \leq n$, té gruix t_i i alçada h_i. Una vegada es decideix quins llibres es fiquen a un prestatge s'ajusta la seva alçada a la del llibre més alt que col·loquem al prestatge. Doneu un algorisme que ens permeti col·locar els n llibres a les prestatgeries de la biblioteca de manera que es minimitzi la suma de les alçades dels prestatges utilitzats.

Sol. Mòltes solucions possibles. La que se indica a l'ajut: construir un nou graf G' idèntic al G, excepte que que per a tota aresta $\overrightarrow{(u,v)}$ definim $w\overrightarrow{(u,v)} = w(v)$. 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 aG' tindrà pes $w\overrightarrow{(u_1, v_1)} + \cdots + w\overrightarrow{(v_k, u_2)}$ 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.

Exercici 4 (3 punts) 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. Cada dispositiu té un abast de transmissió limitat, pot comunicar-se amb altres dispositius que es troben com a màxim a r metres d'ell.

Ens agradaria poder determinar si hi han suficients camins que no comparteixen arcs entre els dispositius de dues zones diferents, A i B, a fi de garantir que un senyal d'emergència pot arribar sense problemes des d'algun dispositiu a A a algun dels dispositius a B. Per tal d'evitar interferències volem, a més, controlar el *nivell de participació* dels dispositius. Aquest nivell és el màxim nombre de camins en què un dispositiu hi participa.

Suposeu que després del llançament podem obtenir les coordenades $p_i = (x_i, y_i)$ dels n dispositius operatius que formen la xarxa inicial. Suposeu també, que les regions A i B són rectangles i que us donen les coordenades de les seves cantonades. Finalment, us donen també un valor c que determina el nombre de camins mínim que voldríem tenir entre A i B.

(a) Dissenyeu un algorisme per determinar si és possible trobar c camins que no comparteixin arestes entre A i B.

Solució: Hemos visto en clase como calcular el máximo número de caminos que no comparten aristas entre dos vertices poniendo capacidad 1 en los arcos. Aquí nos piden calcular este número pero entre dos conjuntos. Para ello añadiremos dos vértices s y t. Conectaremos s con todos los dispositivos en la zona A y conectaremos todos los dispositivos de la zona B con t. Añadimos un arco (a, b) si el dispositivo b está dentro del rango de a. Obtenemos una red de flujo adecuada añadiendo capacidad 1 a todos las conexiones entre dispositivos, ya que solo se pueden utilizar una vez. Las conexiones desde s o hacia t se pueden usar más veces, ya que corresponden con pasos por uno de los vértices, pero estos no tienen límite. Podemos asignarles capacidad ∞.

Una vez obtenido el flujo con valor máximo comprobamos que su valor sea o no mayor o igual que c.

Si utilizamos Ford-Fulkerson, podemos para la ejecución la primera vez que el valor del flujo sea $\geq c$. Esto nos da coste $O(cn)$. Si utilizamos Edmons-Karp el coste es $O(n^2m)$. Como el número total de caminos puede ser exponencial c puede ser un número muy grande. Para garantizar coste polinómico usaremos el segundo análisis de coste.

(b) Donat $d > 0$, dissenyeu un algorisme per determinar si hi han c camins que no comparteixen arestes entre A i B de manera que el nivell de participació de cada node és, com a molt, d.

Solució: Para limitar el número de veces que pasamos por un vértice modificamos la red del paso anterior. Ponemos dos nodos u_1, u_2 por cada dispositivo u, y el arco (u_1, u_2) con capacidad d. Por cada arco (u, v) colocamos el arco (u_2, v_1). Así cada camino que pase por u en el grafo anterior contiene el arco (u_1, u_2). Si el max flow de esta red es $\geq c$ la respuesta es sí. El coste es el mismo que en el caso anterior.

(c) Dissenyeu un algorisme per determinar el mínim nivell de participació que permeti trobar c camins que no comparteixin arestes entre A i B amb aquest nivell màxim de participació dels nodes.

Solució: Si consideramos c caminos e nivel de participación es como mucho c. Podemos hacer una búsqueda dicotómica para determianar el valor d más pequeño que nos permite obtener c caminos. Esto nos da $\log c$ ejecuciones del algoritmo del apartado anterior. Es decir $O(\log c\, n^2 m)$.