Cognoms, Nom DNI

Algorísmia QT 2017–2018

Examen final **15 de Gener de 2018**

Durada: 2h 50m

Instruccions generals:

- L'exercici 1 s'ha de resoldre fent servir l'espai reservat per a cada resposta.
- Heu d'argumentar la correctesa i 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 la solució de cadascun dels exercicis.
- La puntuació total d'aquest examen és de **6 punts** (nota E).

Exercici 1 (3 punts)

(a) (0.3 punts) Utilitzant l'algorisme d'ordenació RADIX es poden ordenar en temps $O(n \lg n \lg n)$, n enters diferents amb valors entre 1 i $n^{\lg n}$.

Solució: Fals, si utilitzem RADIX amb base n, aleshores cada enter nombre te $d = \log_n n^{\lg n} = \lg n$ dígits, per tant el temps de computació es $\Theta(nd) = \Theta(n \lg n)$

(b) (0.4 punts) Donat un vector A amb n elements, és possible posar en ordre creixent els $\sqrt{n}$ elements més petits i fer-ho en $O(n)$ passos?

Solució: Sececcionar el element $\sqrt{n}$-èsim i particionar al voltant, d'aquest element (cost $O(n)$). Ordenar la part esquerra en $O(\sqrt{n}^2)$.
Alternativament, construir un min-heap en $O(n)$ i extreure el mínim element $\sqrt{n}$ cops, el nombre de passos és $O(n + \sqrt{n} \lg n)$.

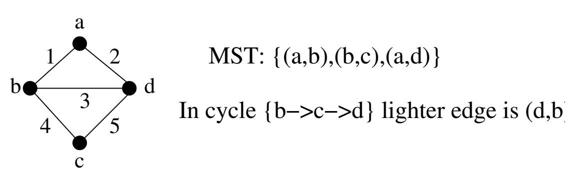
(c) (0.3 punts) Antoni escull un enter entre 1 i 10^6, Bartomeu ha d'endevinar quin és l'enter escollit fent el mínim nombre de preguntes que tinguin com a resposta (sí/no). Doneu una fita superior a aquest mínim nombre de preguntes.

Solució: Utilitzant l'arbre de comparació explicat a classe, necessitem com a màxim la longitut de la bramca més llarga, $\lceil \lg(10^6) \rceil = \lceil 6 \lg 10 \rceil = 24$

(d) (0.3 punts) És cert que si a una xarxa $(s - t)$ totes les capacitats són racionals aleshores el flux màxim també serà un racional? (recordeu que un nombre racional és el quocient de dos enters, per ex, 3/4 o 7/5)

Solució: Cert. Agafem el mínim comú múltiple dels denominadors de totes les capacitats, sigui m, multipliqueu cada capacitat per m això transforma la xarxa en una xarxa amb on les capacitat són enters. Trobeu el valor d'un flux màxim, dividiu per m això dona el valor del flux màxim de la xarxa original, que serà un racional. (Tot enter també és racional, ja que es pot expressar com a quocient de dos enters)

(e) (0.3 punts) Sigui $G = (V, E)$ un graf amb pes $w : E \to \mathbb{Z}^+$, aleshores l'aresta amb menys pes a un cicle ha de ser al MST, i la aresta de més pes al cicle NO ha de ser al MST. Digueu si aquesta afirmació és certa o falsa. **Solució:** Tal com està escrit és fals, ja que és cert que l'aresta amb més pes ho pot pertany al MST, però com la figura indica, no sempre és cert respecte a l'aresta amb menys pes.



(f) (0.3 punts) Suposem que volem el canvi de n cèntims, utilitzant el mínim de monedes de denominacions 1, 10 i 25 cèntims. Considereu la següent estratègia golafre: si la quantitat que em queda és m; agafem la moneda més gran que no sigui més gran que m; restem el valor d'aquesta moneda de m, i repetim.

Aquesta estratègia resol el problema plantejat?

Solució: No, tal i com es dedueix del següent contraexemple. Quan $n = 30$ podem donar canvi amb 3 monedes de 10 cèntims. L'algorisme proposat agafaria una moneda de 25 cèntims, i després 5 de 1 cèntim. L'algorisme fa servir 6 monedes en comptes de les 3 de la solució òptima.

(g) (0.3 punts) Recordeu el problema del recobriment de vèrtexs amb k-centres a $\mathbb{R}^2$: *Donat com entrada un conjunt de punts a $\mathbb{R}^2$, $X = \{x_1, \dots, x_n\}$, on els punts venen representats per les seves coordenades cartesianes, donada una taula amb totes les $\frac{n(n-1)}{2}$ distancies Euclidianes $D = \{d(x_i, x_j)\}$, i donat un enter $k < n$, seleccionar un conjunt de k centres $C \subseteq X$ i el mínim radi r tal que els k cercles centrats al punts a C siguen un recobriment d'X.* Vam comentar a classe que el problema era NP-complet per $k > 1$. Doneu un algorisme determinista que resolgui exactament el problema del k-centre per el cas de que $k = 1$, i ho faci en temps polinòmic.

Solució: Si $k = 1$ només hem de cercar el punt per el que el punt mes llunyà estigui ho més a prop possible. Per fer ho s'ha de recorre la taula ($\Theta(n^2)$) calculant la distancia màxima per cada fila. El mínim d'aquestes distancies es el valor r que ens demanen. L'algorisme té cost $O(n^2)$.

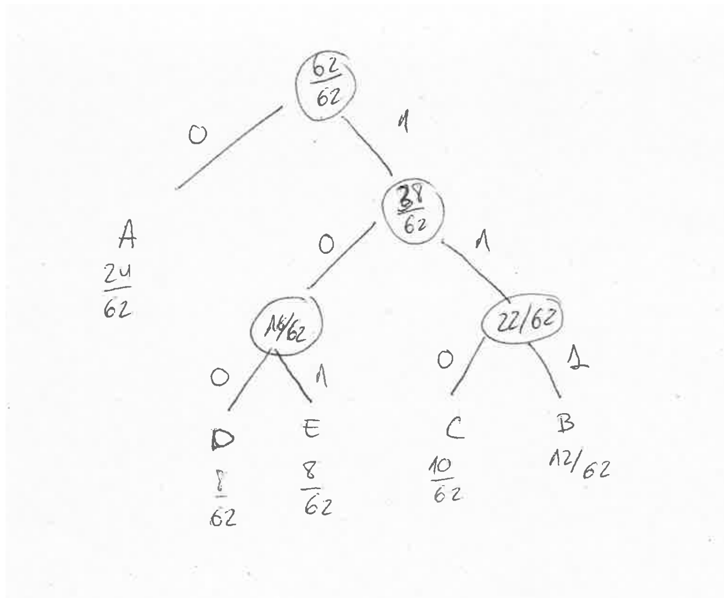
(h) (0.3 punts) Tenim una taula de hash T, amb grandària $2n$, on resolem les col·lisions amb llista d'adjacència. Suposem que introduïm $n/2$ claus a la taula T utilitzant una funció d'hashing que és simple uniforme. Es cert que el nombre esperat de claus per a cada registre és $1/4$?

Solució: Cert. Definim una variable aleatòria indicadora X_i ($1 \leq i \leq n/2$) que és $X_i = 1$ si la clau i va al registre $T[j]$, altrament $X_i = 0$. Aleshores, $\mathbf{E}[X_i] = \mathbf{Pr}[X_i = 1] = \frac{1}{2n}$. Per tant el nombre experat d'elements que va a $T[j]$ és $\mathbf{E}\left[\sum_{i=1}^{n/2} X_i\right] = \sum_{i=1}^{n/2} \mathbf{E}[X_i] = \frac{n}{4n} = \frac{1}{4}$.

(i) (0.5 punts) Considerem una cadena amb longitud 62, formada amb l'alfabet $\Sigma = \{A, B, C, D, E\}$. Sabem que els cops que apareix cada lletra són: $A = 24, B = 12, C = 10, D = 8, E = 8$. Noteu que si comprimim la cadena, utilitzant un mètode de compressió amb longitud fixada, cada lletra necessitarà 3 bits i la longitud de la cadena comprimida és 186 bits. Per a comprimir més eficientment utilitzem el mètode de Huffman:

i. (0.1 punts) Quina tècnica algorísmica utilitza Huffman?
(**Sol.** Greedy)

ii. (0.3 punts) Dibuixa l'arbre de Huffman.



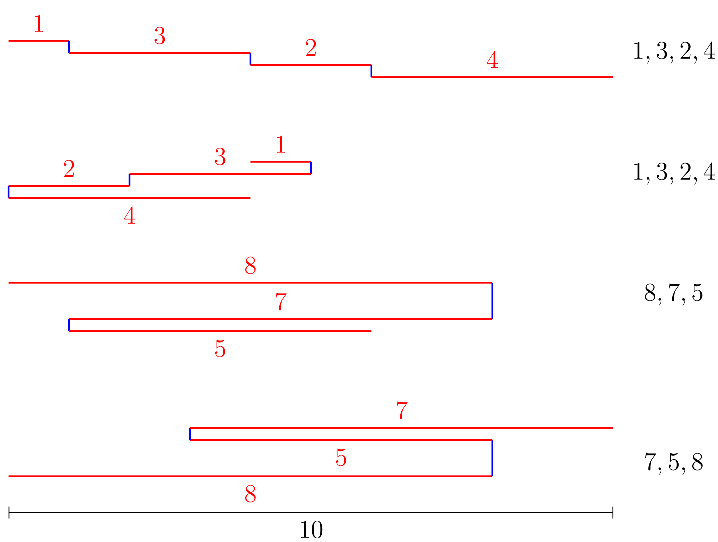
iii. (0.1 punts) Quina es la longitud total de la cadena comprimida amb Huffman?

Sol. $A \to 0$, $B \to 111$, $C \to 110$, $D \to 100$, $E \to 101$ i ens dona
$24 \times 1 + 12 \times 3 + 10 \times 3 + 8 \times 3 + 8 \times 3 = 138$.

Exercici 2 (1.5 punts) Un metre de fuster (com el de la figura de sota) està format per uns quants segments de fusta habitualment iguals. Cada segment és rígid i s'uneix al previ i/o al següent pels extrems de manera que es pot rotar completament a les unions.

En aquest problema considerarem una generalització de metre de fuster en el que els segments poden tenir longituds diferents encara que tots tenen la mateixa amplada. A més cada segment té com a molt 100cm de llargada. Així un metre de fuster està format per n segments de llargades $l_1, \ldots, l_n$ (en aquest ordre) on totes les longituds dels segments son enters al interval $[0, 100]$. Per simplificar la notació considerarem també els extrems $A_0, \ldots A_n$, on A_0 és l'extrem lliure del primer segment, A_1 és l'extrem comú al primer i segon segment, etc, i A_n és l'extrem lliure del segment n-ésim.

Volem analitzar el problema de plegar el metre per tal de ficar-lo a dintre d'una caixa. Per exemple, si els segments són de longitud 1, 3, 2 i 4, el metre es pot guardar en una caixa de longitud 10 (plegar a l'interval [0,10]), però podem fer-ho també en una caixa de longitud 5 (a l'interval [0,5]). Si els segments tenen longitud 8, 7 i 5 en aquest ordre, es pot guardar plegat en una caixa de 8, però si els segments son 7, 5 i 8, llavors la caixa més petita en la que es pot plegar té longitud 10 metres. A la figura de sota teniu una representació estilitzada i bidimensional d'aquests plegaments.



(a) (0.5 punts) Considereu l'algorisme següent

procedure FOLD INSIDE INTERVAL($L(n)$)
    Let $m = \max L[i]$
    Place A_0 at position 0
    for $i = 1, \ldots, n$ **do**
        if it is possible to place A_i to the left of A_{i-1} inside $[0, 2m]$ **then**
            place A_i to the left of A_{i-1}
        else
            place A_i to the right of A_{i-1}
        end if
    end for
end procedure

Demostreu que Fold inside interval determina un plegat que ens permet ficar el metre dintre de l'interval $[0, 2m]$ on m es la longitud del segment més llarg i analitzeu-ne el seu cost.

Sol. Solo necesitamos comprobar que cuando colocamos A_i a la izquierda de A_{i-1} la posición de A_i está dentro de $[0, 2m]$. En este caso sabemos que A_i no se puede colocar a la derecha, por lo tanto si $j \in [0, 2m]$ es la posición de A_{i-1}, tenemos que $j + L[i] > 2m$, como $L[i] \leq m$, tenemos que $j > m$. Por lo tanto $j - L[i] > 0$. Por lo tanto el plegado nos permite colocar el metro en $[0, 2m]$.

(b) (0.5 punts) Considereu el problema Min-Fold: Donat un metre de fuster, format per n segments de llargades $l_1, \ldots, l_n \in \mathbb{N}$ (en aquest ordre), $0 < l_i \leq 100$, trobar la llargada ℓ més petita que ens permeti ficar el metre dintre de l'interval $[0, \ell]$.

Dissenyeu un algorisme que ens permeti resoldre Min-Fold i analitzeu-ne el seu cost temporal i espacial.

Ajut: Penseu en com resoldre recursivament el problema de determinar si un metre de fuster es pot ficar a dintre de l'interval $[0, k]$ posant-hi l'extrem A_0 a la posició $j \in [0, k]$, per valors raonables de k.

Sol. Voy a utilizar una variación recursiva del algoritmo FOLD INSIDE INTERVAL par resolver Min-Fold. Para un valor de k fijado, el algoritmo resolvera recursivamente el problema de determinar si se puede o no plegar el metro $L_i, \ldots, L_n$ dentro de $[0, k]$ bajo la condició de que A_i se ubique en la posició $j \in \{0, \ldots, k\}$.

procedure FOLD INSIDE INTERVAL REC(i, j)
    Place A_i at position j
    Left = Right = False
    if $j - L[i] \geq 0$ **then**
        (it is possible to place A_{i+1} to the left of A_i inside $[0, k]$)
        Left = FOLD INSIDE INTERVAL REC($i + 1, j - L[i]$)
    end if

```
    if j + L[i] ≤ k then
        (it is possible to place A_{i+1} to the right of A_i inside [0, k])
        Right = Fold inside interval rec(i + 1, j + L[i])
    end if
    return (Left or Right)
end procedure
```

Como una vez hemos ubicado A_i en una posición j, el punto A_{i+1} solo puede ubicarse a la derecha o a la izquierda de j, el algoritmo explora todas las posibilidades y por ello es correcto. El algoritmo solo tiene dos parámetros i, $0 \leq i \leq n$, y j, $0 \leq j \leq k$. Por lo que el número de subproblemas es nk. El costo implementándolo con meomización o con tabla será $O(nk)$.

Para determinar si el metro se puede plegar en $[0, k]$ tendríamos que ver si para algún valor de $j \in [0, k]$ Fold inside interval rec$(1, j)$ devuelve cierto. Tendremos un coste adicional $O(k)$.

Finalmente, para resolver Min-Fold, tendriamos que calcular el menor valor k^* para el que el metro se puede plegar en $[0, k^*]$. Por el apartado (a) sabemos que $k^* \leq 2m$. Podemos implementar una búsqueda dicotómica usando el algoritmo previo. El coste total es $O(nm \log m)$. Teniendo en cuenta el enunciado, $m \leq 100$, por lo que el coste del algoritmo es $O(n)$.

(c) (0.25 punts) Analitza el cost de l'algorisme proposat a l'apartat (b) en el cas que els segments del metre de fuster poden tenir qualsevol longitud.

Sol. Si no tenemos el limite de 100 el coste del algoritmo es $O(nm \log m)$. Como m es un número que es parte de la entrada el coste es pseudopolinómico, y por tanto tiene coste exponencial en el tamaño de la entrada.

(d) (0.25 punts) És Fold inside interval una 2-aproximació a Min-Fold?

Sol. Teniendo en cuenta que m es el segmento de longitud máxima, cualquier plegado en $[0, k]$ requiere que $k \geq m$. En particular la solución optima en $[0, k^*]$ tiene que cumplir $k^* \geq m$ y como la que obtenemos en (a) es $2m$, $2m \leq 2k^*$ con lo que podemos concluir que Fold inside interval es una 2-aproximación.

Exercici 3 (1.5 punts) El Rector de la UPC de cara a fomentar la germanor entre el personal de la UPC ha decidit establir activitats mensuals per incentivar la interacció entre els diferents estaments de PDI i PAS. Les festes se celebraran normalment un cop al mes. Suposem que hi han k grups en el personal, $C_1, \ldots, C_k$, no necessàriament disjunts. El rectorat produeix una llista L d'activitats que tindran lloc en el curs acadèmic, i per a cada activitat $a \in L$, hi ha un nombre màxim $M(a)$ i un nombre mínim $m(a)$ de gent que pot assistir a a. A més, com que els grups tenen diferents nivells d'influència, el rectorat estima, per a cada j, el nombre mínim $s(j)$ de persones del grup C_j que s'han de convidar a cada activitat. Per una altra banda és ben conegut que alguns membres de la UPC no poden participar sovint en activitats lúdiques. Tenint en compte aquest aspecte el rectorat ha determinat per cada membre i del personal de la UPC un valor $t(i)$ indicant el nombre màxim d'activitats a les què i pot participar al llarg del curs. El problema consisteix en, en cas que suigui possible, decidir a qui convidar a cada activitat de manera que es compleixin les restriccions prèvies, o sigui:

- el nombre d'assistents a l'activitat a és $\leq M(a)$ i $\geq m(a)$,
- per a cada grup j hi ha d'haver com a mínim $s(j)$ persones assistint a cada activitat.
- la persona i mai es convidada a més de $t(i)$ activitats.

(a) (0.25 punts) Indiqueu com modelaríeu aquest problema com un problema de programació entera. Una solució factible és el conjunt de invitacions que satisfan les restriccions, i volem minimitzar el nombre total d'invitavions.

Sol. El enunciado no deja claro si las invitaciones las envía la universidad a las personas o a los grupos. En el primer caso una persona recibiría como máximo una invitación por actividad pero no quedaría claro a qué grupo representa. Por otra parte en el segundo potencialmente podría recibir más de una invitación para representar a más de un grupo. En mi formalización asumo que una persona en una actividad solo puede representar a un grupo.

Utilizo una variable x_{ija}, $1 \leq i \leq n, 1 \leq j \leq k, 1 \leq a \leq m$, donde $m = |L|$, indicado que la persona i recibe una invitación a la actividad a como miembro del grupo j. Las tres condiciones del enunciado, junto con la adicional de que una persona solo puede usar una invitación de un grupo nos dan el siguiente program ade programación entera.

$$\min \sum_{i,j,a} x_{i,j,a}$$

$$\sum_{j=1}^{k} x_{i,j,a} = 1 \qquad 1 \leq i \leq n, 1 \leq a \leq m$$

$$\sum_{i=1}^{m} \sum_{j=1}^{k} x_{i,j,a} \leq M(a) \qquad 1 \leq a \leq m$$

$$\sum_{i=1}^{m} \sum_{j=1}^{k} x_{i,j,a} \geq m(a) \qquad 1 \leq a \leq m$$

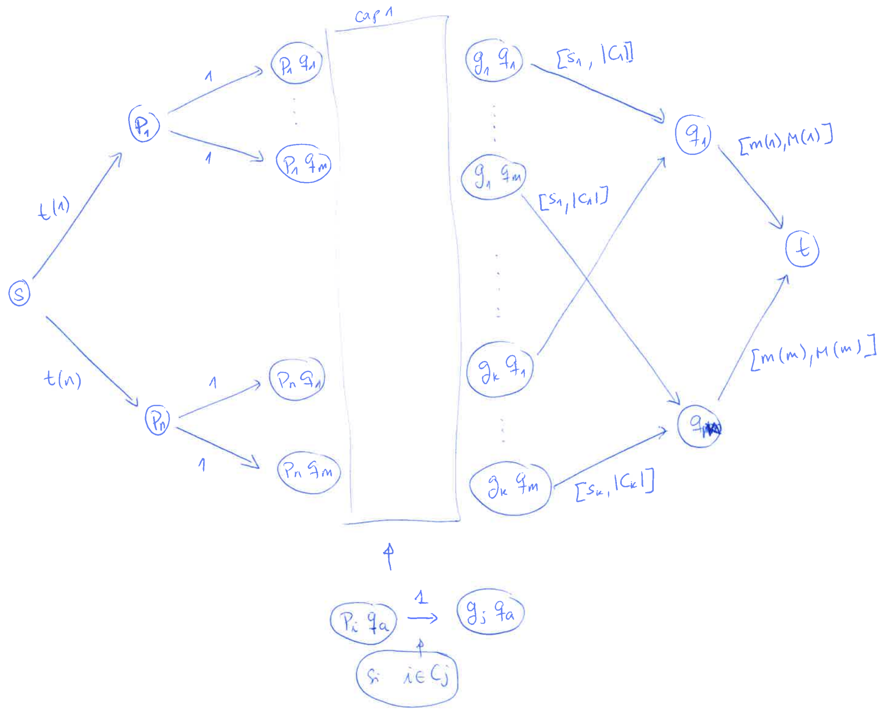
$$\sum_{i=1}^{k} x_{i,j,a} \geq s(j) \qquad 1 \leq j \leq k, 1 \leq a \leq m$$

$$\sum_{a=1}^{m} \sum_{j=1}^{k} x_{i,j,a} = 1 \qquad 1 \leq i \leq n, 1 \leq a \leq m$$

$$x_{ija} \in \{0,1\} \qquad 1 \leq i \leq n, 1 \leq j \leq k, 1 \leq a \leq m$$

(b) (1 punt) Com resoldríeu aquest problema utilitzant un model de xarxa de fluxos?

Sol. Lo planteo como un problema de red de flujo con cotas inferiores en las capacidades de las aristas. La red tiene un conjunto de n nodos $p_1, \ldots, p_n$ (personas), nm nodos $p_i q_a$ (pares persona-actividad), km nodos $g_i q_a$ (pares grupo-actividad), m nodos $q_1, \ldots, q_m$ (actividades). Las conexiones y cotas inferiores y superiores son las siguientes



Observemos que un camino de s a t es de la forma $s \to p_i \to p_i q_a \to g_j q_a \to g_a \to s$ llevando flujo 1 representa que $x_{ija} = 1$. Si tenemos una solución factible del problema de programación entera obtenemos un flujo en la red utilizando los caminos asociados que verifica las restricciones y viceversa.

(c) (0.25 punts) Compareu la complexitat de resoldre el problema en cadascun dels dos models.

Sol. El problema de Programación entera es NP-hard mientras que resolver un problema de redes de flujo, incluso con cotas superiores e inferiores, se puede hacer en tiempo polinómico.