Cognoms, Nom DNI

Algorísmia QP 2016–2017

Examen final **9 de Juny de 2017**

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.
- 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)

Digueu i **justifiqueu** si cada una de les següents afirmacions és certa o falsa. (**La part important és la justificació de la resposta**.)

(a) (0.2) Donat un graf no dirigit $G = (V, E)$, amb pesos $w : E \to \mathbb{R}^+$ i tal que tots els pesos són diferents, l'aresta amb més pes no pot pertanyer a cap MST.

(b) (0.2) És cert que, en general, donat un problema no és fàcil demostrar una fita inferior a la seva complexitat determinista? Pots donar un exemple d'un problema específic on es conegui una fita inferior a la seva complexitat?

(c) (0.2) Considereu un graf $G = (V, E)$ amb pesos sobre les arestes $w : E \to \mathbb{R}$, on tots els pesos són diferents. Si s, t són dos vèrtexs a G, la distancia mínima de s a t és única.

(d) (0.2) El temps per a resoldre un problema de programació dinàmica sempre és $\Theta(k)$, on k és el nombre de subproblemes diferents del problema.

(e) (0.25) Considereu un digraf $G = (V, E)$ amb pesos sobre les arestes $w : E \to \mathbb{R}$, sense cicles positius i amb un vèrtex distingit $s \in V$. Existeix un algorisme eficient per a trobar el camí amb **pes màxim** entre s i qualsevol altre vèrtex $u \in V - \{s\}$. Si existeix, quina és la complexitat del vostre algorisme?

(f) (0.25) Suposem que una taula de dispersió (hashing) amb grandària m conté una única clau k i la resta dels registres són buits. Si tenim una funció de dispersió h que té la propietat de ser *simple uniforme* i apliquem r vegades la funció h a un conjunt de claus d'entrada, la probabilitat de tenir una col.lisió (que una de les r vegades h vagi a parar al registre ocupat per k) és r/m.

(g) (0.25) Sigui X una variable aleatòria indicadora tal que $\mathbf{E}[X] = 1/2$. Aleshores tenim que $\mathbf{E}\left[\sqrt{X}\right] = 1/\sqrt{2}$.

Doneu una explicació **breu i concisa**.

(i) (0.25) Què és un arbre de Merkle? Exactament, quin és l'avantatge de tenir un arbre de Merkle sobre una altra estructura de dades cóm un arbre binari equilibrat?

(j) (0.2) Donat un text T, suposem que els símbols a, b, c, d, e apareixen amb una freqüència $1/2, 1/4, 1/8, 1/16, 1/16$

i. Quina és la compressió de Huffman per a cada símbol

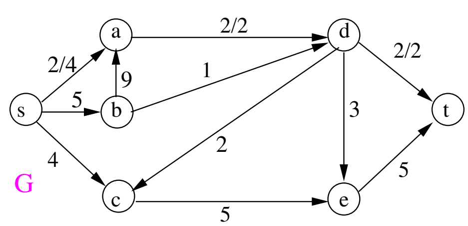
ii. Si T té 1000000 de caràcters, quina és la longitud de la compressió de T (utilitzant Huffmann)?

(k) (0.25) Considereu el problema de la selecció d'activitats per a un processador, quan l'activitat té pesos: Donat un conjunt d'activitats $S = \{1, 2, \ldots, n\}$ on l'activitat i ve donada per un temps de començament s_i, un temps de finalització f_i i un pes w_i, volem trobar un subconjunt d'activitats que maximitzin la suma dels pesos, i.e. $\sum_{i \in S} w_i$. Considereu el següent algorisme voraç: Ordeneu incrementalment les tasques per temps de finalització (llista A). Torneu a ordenar pel pes, de més gran a més petit (llista B). Utilitzant la llista B, anem escollint les tasques amb pes w més gran, utilitzeu la llista A per assegurar que les tasques seleccionades són compatibles. Argumenteu la correctesa d'aquest algorisme voraç. Doneu la seva complexitat, assumint que la entrada no està ordenada de cap manera.

(l) (0.5) Donada una taula $A[1\cdots 2n+1]$ direm que és oscil.lant si per $1 < i < 2n+1$, si i és parell tenim $A[i-1] \geq A[i]$ i $A[i] \leq A[i+1]$, i si i és senar tenim $A[i-1] \leq A[i]$ i $A[i] \geq A[i+1]$, amb $A[1] \geq A[2]$ i $A[2n] \leq A[2n+1]$. Per exemple, $14, 7, 14, 2, 21, 3, 18, 2, 17$ és una taula oscil·lant amb longitud 9. Donada com a entrada una taula no ordenada d'enters $B[1\cdots 2n+1]$ descriviu un algorisme lineal que transformi B en una taula oscil·lant.

.

(m) (0.25) Considereu la següent xarxa $G = (V, \vec{E}, s, t, c)$ on $c : \vec{E} \to \mathbb{Z}^+ \cup \{0\}$ amb un flux f amb valor 2 com s'indica a la figura de sota. Expliqueu com serà la següent iteració de l'algorisme de Edmonds-Karp, Dinic, per trobar un flux amb un increment net de valor. Dibuixeu el graf residual G_f, el graf de nivells L_{G_f} i digueu quin serà el camí augmentador que incrementa el valor del flux. Quin és el valor del nou flux f'? Quina és la complexitat d'aquesta iteració?



Exercici 2 (1.5 punts) (**L'alineament de seqüències**). Quan es descobreix un nou gen, una manera estàndard de descobrir la seva funció és mirar a una base de dades de gens que ja s'han estudiat molt i per als què es coneix perfectament el què fan, i trobar coincidències el més ajustades que sigui possible entre el nou gen i algun dels gens coneguts. La proximitat de dos gens es mesura pel grau d'alineació. Per formalitzar això en termes computacionals, podem pensar que un gen és una cadena sobre un alfabet $\Sigma = \{A, G, C, T\}$. Considerem dos gens $x = ATGC$ i $y = TACGCA$, una alineació de x i y és una manera de fer coincidir aquestes dues cadenes escrivint-los en columnes, per exemple:

$$\begin{array}{cccccc} - & \mathrm{A} & \mathrm{T} & \mathrm{G} & \mathrm{C} & - \\ \mathrm{T} & \mathrm{A} & \mathrm{C} & \mathrm{G} & \mathrm{C} & \mathrm{A} \end{array}$$

on el caràcter $-$ denota un forat o una no-coincidència. Els caràcters a cada cadena han d'aparèixer en ordre, i cada columna ha de contenir un caràcter d'almenys d'una de les cadenes.

La qualitat d'una alineació s'especifica utilitzant una matriu de puntuació δ amb dimensió 5×5. A l'exemple previ, l'alineació té una puntuació resultant de:

$$\delta(-, T) + \delta(A, A) + \delta(T, C) + \delta(G, G) + \delta(C, C) + \delta(-, A).$$

L'elecció dels valors de δ no és senzilla i depèn de l'aplicació concreta. N exemple de matriu de puntuacions és següent:

$$\begin{array}{cc} \begin{array}{ccccc} A & T & C & G & - \end{array} & \\ \begin{pmatrix} +1 & -1 & -1 & -1 & 0 \\ -1 & +1 & -1 & -1 & 0 \\ -1 & -1 & +1 & -1 & 0 \\ -1 & -1 & -1 & +1 & 0 \\ 0 & 0 & 0 & 0 & \end{pmatrix} & \begin{array}{c} A \\ T \\ C \\ G \\ - \end{array} \end{array}$$

Segons aquesta matriu de puntuació l'alineació proposta tindria una puntuació de 2.

Doneu un algorisme que prengui com a entrada dues cadenes X i Y d'ADN, amb longitud n i m respectivament ($X[1, \ldots, n], Y[1, \ldots m]$) i una matriu de puntuació δ, i retorni l'alineació amb puntuació més alta. El temps d'execució del vostre algorisme ha de ser $O(mn)$.

Exercici 3 (1.5 punts) **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 encàrrec 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 $s, t \in V$ 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 que es pugui, el nivell de seguretat que es pot assolir. Per aquesta raó 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 una de les 5 nits com a molt.

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 3: 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, aquella nit poden circular per ell fins a tres camions carregats.

Proporcioneu algorismes per a cada escenari, tals que:

(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: (1) cada camió carregat fa com a molt un trajecte per nit; (2) globalment es compleixen les restriccions imposades per l'escenari; i (3) el nombre total de trasllats es més gran que $\frac{2}{3}ck$.