Cognoms, Nom DNI

Algorísmia QP 2016–2017

Una Possible Solució a l'Examen final del 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.

Solució: FALS: Considerem por exemple el cas en que G és un arbre.

(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?.

Solució CERT. Pel quantificador universal: $\forall$ entrada **no existeix cap algorisme**. Si, ordenar n claus utilitzant utilitzant comparació és $\Omega(n \lg n)$

(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.
Solució: CERT. Si es pot definir la distancia (no hi han cicles amb pes negatiu) aquesta 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.
Solució: FALS. Hem de multiplicar per el temps per a resoldre cada subproblema diferent.

(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?
Solució: CERT: Construïu G' on el unic canvi consisteix en negar $(-w)$ el pes de cada aresta. Aleshores podeu aplicar Bellman-Ford-Moore per a trobar el camí mínim $s \to u$ a G', aquest camí és un camí amb pes màxim al graf original G. $O(nm)$.

(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.
Solució: FALS, la probabilitat de que una de les r aplicacions de h vagi a parar al registre que conté k és 1- la probabilitat de que cap de les r aplicacions vagi a parar al registre que conté k, per tant la probabilitat que volem és $1 - (1 - \frac{1}{m})^r$.

(g) (0.25) Sigui X una variable aleatòria indicadora tal que $\mathbf{E}[X] = 1/2$. Aleshores tenim que $\mathbf{E}\left[\sqrt{X}\right] = 1/\sqrt{2}$.
Solució: FALS. Com X és indicadora tenim que $X = 0$ o $X = 1$, per als dos possibles valors tenim que $\sqrt{X} = X$. Per tant, $\mathbf{E}\left[\sqrt{X}\right] = \mathbf{E}[X] = 1/2$.

Doneu una explicació breu i concisa

(i) (0.25) Què és un arbre de Merkle? Exactament, quina és l'avantatge de tenir un arbre de Merkle sobre una altra estructura de dades cóm un arbre binari equilibrat?
Solució Mirar les transparències de classe

(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)?

Solució:

i. una possible solució:$a = 0, b = 10, c = 110, d = 1110, e = 1111$

ii. $1000000 \times (1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 4 \times \frac{1}{16} + 4 \times \frac{1}{16}) = 875000$

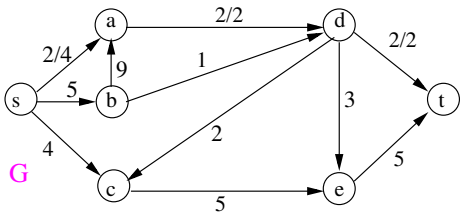
(k) (0.25) Considereu el problema de la selecció d'activitats per a un processador, quan l'activitat té pesos: Donat un conjunt d'activitats $S = \{1, 2, \ldots, n\}$ on l'activitat i ve donada per un temps de començament s_i, un temps de finalització f_i i un pes w_i, volem trobar un subconjunt d'activitats 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.

Solució: No, necessitem fer ho utilitzant PD. EL contraexemple és a les transparències de classe.

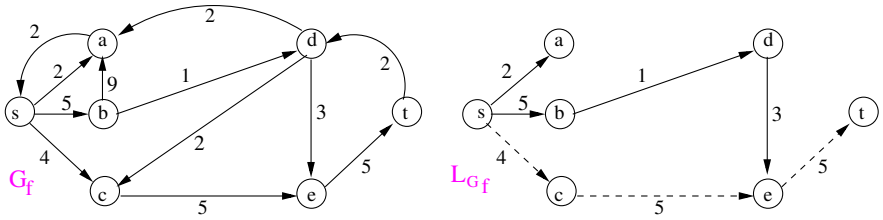
(l) (0.5) Donada una taula $A[1 \cdots 2n+1]$ direm que és oscil.lant si per $1 < i < 2n+1$, si i és parell tenim $A[i-1] \geq A[i]$ i $A[i] \leq A[i+1]$, i si i és senar tenim $A[i-1] \leq A[i]$ i $A[i] \geq A[i+1]$, amb $A[1] \geq A[2]$ i $A[2n] \leq A[2n+1]$. Per exemple, $14, 7, 14, 2, 21, 3, 18, 2, 17$ 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.

Solució: Una forma de resoldre el problema es garantir que els valors a posicions parells siguin menors que els valors a posicions senars. Calculem la mitjana en temps lineal i després. Els valors menors o igual que la mitjana aniran a les posicions parells i la resta a les senars.

(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?



Solució:



A L_{G_f} podem escollir el camí augmentador $s \to c \to e \to t$ que ens permet incrementar f fins a $f' = f + 4 = 6$. (Encara es necessitaria un iteració més per a obtenir el flux màxim.) El graf L_{G_f} assegura que a cada iteració escollim el camí $s \rightsquigarrow t$ amb menys arestes. Si representem els digrafs per llistes d'adjacència, el cost de modificar $G \to G_f \to L_{G_f}$ és $O(m)$ aplicar BFS a L_{G_f} és $O(m+n)$ el que dona un cost per iteració de $O(m+n)$

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 $x = ATGCC$ i $y = TACGCA$, una alineació de x i y és una manera de fer coincidir aquestes dues cadenes escrivint-los en columnes, per exemple:

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

on el caràcter $-$ denota un forat o una no-coincidència. Els caràcters a cada cadena han d'aparèixer en ordre, i cada columna han 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:

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

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 $O(mn)$.

Solució: Molt semblant a la distancia d'edició feta a classe, però utilitzant la següent recurrència

$$E[i, j] = \max\{E[i-1, j] + \delta(x_i, -), E[i-1, j-1] + \delta(x_i, y_j), E[i, j-1] + \delta(-, y_j),$$

amb condicions inicials $\forall i, j > 0$, $E[0, 0] = 0$; $E[i, 0] = E[i-1, 0] + \delta(x_i, -)$; i $E[0, j] = E[0, j-1] + \delta(-, y_j)$. Com amb el problema de la distància d'edició, el temps total sera $O(nm)$.

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 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.