Algorísmia QT 2018–2019

Examen parcial **14 de Novembre de 2018**

Durada: 1h 50m

Instruccions generals:

- L'exercici 3 s'ha 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, Ex 2 i Ex 3).
- La puntuació total d'aquest examen és de **10 punts**.

Exercici 1 (3 punts) (Video blocs)

Tenim n blocs de vídeo que s'han d'enviar per una única línia de comunicació a mida que es filmen. El bloc i-ésim té una grandària de b_i bits i està disponible per iniciar la seva transmissió a temps s_i, on s_i i b_i són enters positius. El temps de transmissió en el canal es proporcional al nombre de bits, podeu assumir que transmetre un bit requereix una unitat de temps.

Encara que no es poden enviar dos paquets al mateix temps, la transmissió d'un bloc de vídeo es pot suspendre a qualsevol instant per completar-la més endavant. Per exemple si tenim dos blocs de vídeo amb $b_1 = 5$, $s_1 = 2$, $b_2 = 3$ i $s_2 = 0$ podríem enviar a temps 0 els 3 bits del segon paquet i a temps 3 el 5 bits del bloc 1 o bé, enviar a temps 0, 2 bits del bloc 2, després els 5 bits del bloc 1 i finalitzar amb el bit restant del bloc 2. També seria possible enviar a temps 0, 2 bits del bloc 2, després 2 bits del bloc 1, finalitzar el bit restant del bloc 2, i finalitzar els bits del bloc 1. Observeu que no podem començar la transmissió del bloc 1 fins al instant 3.

Una vegada fixada la planificació de l'ordre en què enviarem els bits dels diferents blocs ens interessa el temps en el que finalitza la transmissió de cada bloc. Anomenem f_i al temps en el que finalitza la transmissió del bloc i. Observeu que, per a l'exemple anterior, la primera planificació dona $f_1 = 8$ i $f_2 = 3$, la segona planificació dona $f_1 = 7$ i $f_2 = 8$ menres que la tercera dona $f_1 = 8$ i $f_2 = 5$.

Volem obtenir una planificació de la transmissió dels n blocs de vídeo que minimitze la suma dels temps de finalitzaciço de la transmissió dels blocs, és a dir $\sum_{i=1}^{n} f_i$. Doneu un algorisme que per a una entrada de n blocs, cadascun especificat per (b_i, s_i), determini una planificació (començant a temps 0) resolgui el problema proposat.

Exercici 2 (2.5 punts) (Tallant ADN).

En termes computacionals, podem pensar que una cadena de ADN es un mot sobre l'alfabet $\{A, C, G, T\}$. El cost d'una cadena de ADN el definim com $\prod_{X \in \{A,C,G,T\}} (\text{freq}[X] + 1)$ on $\text{freq}[X]$ es el nombre d'ocurrències del símbol X. Per exemple, si la cadena es $ACGAC$ el seu cost es $3 \times 3 \times 2 \times 1 = 18$

Tenim una cadena de ADN de llargada n i volem dividir-la en k trossos (subcadenes no buides), de manera que es minimitze el cost màxim d'entre les k portions. Per exemple per $ACGAC$ i $k = 2$ tenim 4 formes de trencar la cadena en dos i una de les que ens dona el millor cost es AC i GAC (amb cost màxim 8). Proporcioneu un algorisme ho més eficient possible per a resoldre aquest problema, donats S i k.

$l = \log_{10}(n^5 - 1)$

Cognoms, Nom Data

Exercici 3 (4.5 punts)

(a) (0.5 punts) Donats n enters amb rang entre $\{0, 1, 2, \ldots n^5 - 1\}$, quin és el temps pitjor de computació per ordenar els n enters **fent servir RADIX**, quan :

i. utilitzem una representació base-10?,

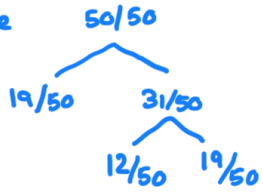
$O((n+b)*l) = O((n+10)*\log_{10}(n^5-1)) = O((n+10)*5\log n) = O(n*\log n)$

ii. utilitzem una representació base-n?.

$O((n+b)*l) = O((n+n)*\log_n(n^5-1)) = O((n+n)*5) = O(10*n) = O(n)$

(b) (0.5 punts) En un codi de Huffman, si tots els caràcters tenen freqüència $< 2/5$, aleshores no hi han caràcters que es pugui codificar amb longitud 1.

Si les freqüències són: $\left(\frac{19}{50}, \frac{19}{50}, \frac{12}{50}\right)$ L'arbre quedaria de la següent manera:

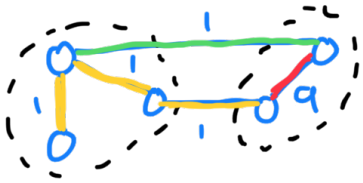


Com podem veure l'arbre té una fulla amb longitud 1.

(c) (1 punt) 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?

(d) (0.5 punts) Donat un graf no dirigit $G = (A, E)$ amb pesos $w : E \to \mathbb{Z}$, raoneu com trobaríeu *l'arbre d'expansió amb màxim pes* of G.

(e) (0.5 punts) Sigui $G = (V, E, w)$ un graf no-dirigit, connectat i amb pesos $w : E \to \mathbb{Z}^+$. Si fem una partició de V en V_1 i V_2, calculem el MST de cada subgraf, i unim amb l'aresta de pes més petit entre V_1 i V_2, el resultat és un MST per G?



Fals

(f) (0.5 punts) Suposem que comencem un procés dinàmic per formar un graf G amb amb un conjunt de n V vèrtexs i no arestes. Sigui $G_0 = (V, \emptyset)$. G_0 te n components connexes. Considerem el procés d'afegir una aresta a cada pas, fins a tenir un graf amb m arestes, per tant tindrem $G_0, G_1, \ldots, G_m$, on G_m serà el graf que volem G, i al pas t, G_t afegirem l'aresta e_t a les que ja havíem afegit prèviament $\{e_1, e_2, \ldots, e_{t-1}\}$. Doneu un algorisme $o((m+n)\log n)$ per a calcular el nombre de components connexes a cada pas del procés.

Utilitzant particions

(g) (1 punt) Recordeu que una formula booleans està donada en *forma normal conjuntiva (CNF)* com una conjunció de clàusules on cada clàusula és una disjunció de literals. En el problema **MaxSat** donada una fórmula booleana en CNF F (amb n variables i m clàusules) volem troba una assignació de valors booleans a les variables que facin cert el major nombre possible de clàusules a F. Considereu l'algorisme següent:

- Obtenir el nombre c_0 de clàusules a F què son cert quan assignem a totes les variables el valor fals.
- Obtenir el nombre c_1 de clàusules a F què son cert quan assignem a totes les variables el valor cert.
- Si $c_0 > c_1$ tornem l'assignació $x_i = 0$, $1 \leq i \leq n$. En cas contrari tornem l'assignació $x_i = 1$, $1 \leq i \leq n$.

Demostreu que aquest algorisme és una 2-approximació per a **MaxSat**.