Algorísmia Q2 2020–2021

Examen Final **14 de juny de 2021**

Durada: 2h 50mn

Instruccions generals:

- Heu de resoldre cada exercici fent servir el full corresponent d'aquest enunciat. No oblideu posar el vostre nom a cada full.
- Entregueu per separat les solucions de cada exercici (Ex 1, Ex 2, Ex 3 i Ex 4).
- Heu d'argumentar la correctesa i l'eficiència dels algorismes que proposeu. Per fer-ho podeu donar una descripció d'alt nivell de l'algorisme suficient per tal que, amb les explicacions i aclariments oportuns, justifiqueu 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. Quan la pregunta sigui una qüestió (Cert o Fals?), heu de dir si l'enunciat és cert o fals abans de justificar la resposta.
- 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.
- La puntuació total d'aquest examen és de **10 punts**.

Exercici 1 (2 punts). Heu de justificar totes les vostres afirmacions, en cas contrari la nota de la pregunta serà 0. Quan la pregunta sigui una qüestió (Cert o Fals?), heu de dir si l'enunciat és cert o fals abans de justificar la resposta.

1.A. **(0.25 punts)** Cert o Fals? Donat un graf connex, no dirigit i amb pesos, que té com a mínim 3 vèrtexs i on tots els pesos són diferents, l'aresta amb més pes mai serà a un MST.

1.B. **(0.5 punts)** Tenim una cadena de DNA sobre un alfabet $\{A, C, G, T\}$ la freqüència de cada caràcter a la cadena és, respectivament, del 30%, 20%, 10% i 40%. Doneu una codificació de Huffman per a cada caràcter.

1.C. **(0.5 punts)** Sigui A una taula que conté n claus, de les quals com a màxim hi ha k claus diferents (no necessàriament enters), on $k \leq \lg n$. Es pot ordenar la taula, mantenint la posició relativa inicial dels elements replicats, en temps $o(n \lg n)$?

1.D. **(0.5 punts)** Tenim un graf dirigit acíclic $G = (V, E)$ amb pesos a les arestes $w : E \to \mathbb{R}$. Donats (G, w) i $u, v \in V$, es pot obtenir un camí de u a v de pes màxim a G en temps $O(|V| + |E|)$?

1.E. **(0.25 punts)** Suposeu que una persona ens presenta una possible solució al problema del flux màxim a una xarxa de flux. Doneu un algorisme que en temps lineal ens determini si la solució proposada té valor màxim.

Exercici 2 (2.5 punts) Sigui X un conjunt de n intervals a la recta real. Una coloració pròpia de X assigna un color a cada interval, de manera que dos intervals que se superposen tenen assignats colors diferents. Descriu i analitza un algorisme golafre eficient per obtenir el mínim nombre de colors necessaris per acolorir (amb una coloració pròpia) un conjunt d'intervals X. Podeu assumir que l'entrada està formada per dos vectors $L[1..n]$ i $R[1..n]$, representant els extrems esquerres (L) i drets (R) dels intervals a X.

Exercici 3 (2.5 punts)

Una operació habitual als laboratoris de biologia molecular és la de trossejar una cadena d'ADN, tallant-la a unes determinades posicions d'interès. En termes computacionals, s'acostuma a representar una cadena d'ADN com un mot sobre l'alfabet $\{A, C, G, T\}$.

Assumim que tallar una cadena d'ADN $c = c_1 \cdots c_n$ de longitud n en dos trossos qualssevol té cost n. Quan parlem de fer un tall a la posició p_i, el símbol d'aquella posició queda com a últim símbol del tros esquerre (és a dir, tallem darrera de p_i).

És fàcil veure que, si es vol fer una seqüència de talls a unes determinades posicions d'interès $p_1, \ldots, p_k$, amb $p_i \in [1, \ldots, n]$, l'ordre en què es facin aquests talls afectarà el cost total de l'operació. Preneu com a exemple el cas d'haver de trencar una cadena de longitud $n = 200$ a les posicions 20, 80, 100 i 150. Si fem els talls d'esquerra a dreta, primer tallem una cadena de 200 (a la posició 20) i ens queden dos trossos, un de 20 i un altre de 180; el segon tall el farem a la posició 80 amb cost 180, etc. El cost total serà $600 = 200 + 180 + 120 + 100$. En canvi, podríem fer el primer tall a la posició 100 (amb cost 200) i després tallar cadascun dels dos trossos ($c_1 \cdots c_{100}$ i $c_{101} \cdots c_{200}$) d'esquerra a dreta. Això ens donaria un cost de $480 = 200 + (100 + 80) + (100)$.

Doneu un algorisme de PD tal que, donada una cadena d'ADN i la seqüència de posicions a l'esquerra de les quals volem tallar la cadena, ens proporcioni la forma menys costosa de tallar-la. El vostre algorisme ha de proporcionar el millor cost possible per realitzar els talls i una estructura de dades que ens permiti esbrinar com dur a terme els talls per tal d'aconseguir el cost òptim.

Exercici 4 (3 punts). El RACC està interessat a trobar un mecanisme per poder suggerir als seus clients rutes alternatives per fugir dels embussaments de trànsit al centre de Barcelona. El Servei Català de Trànsit (SCT) els hi ha proporcionat un model simplificat del centre de Barcelona, que està format per un graf $G = (V, E)$ amb un vèrtex per cada cruïlla, i una aresta per cada parell de cruïlles connectades perquè hi ha un carrer que les uneix. A més coneixen el subconjunt de cruïlles $B \subseteq V$ que donen accés a carrers que surten del centre de la ciutat.

Dels telèfons mòbils dels clients, el RACC pot obtenir informació sobre el conjunt $C \subseteq V$ format per les cruïlles més rellevants als llocs on es troben els cotxes dels seus clients. L'SCT proporciona també el conjunt de trams de carrers $E' \subseteq E$ amb un nivell baix de saturació de trànsit i el conjunt $D \subseteq V$ de cruïlles crítiques que vol evitar.

El RACC vol determinar si és possible trobar un conjunt de camins disjunts (que no comparteixen arestes) que faci servir només trams amb baix nivell de saturació tals que, per cada cruïlla $c \in C$ tinguem un camí que comença a c i que acaba a alguna de les cruïlles de B. A més cal que una cruïlla crítica (de D) aparegui com a molt a 2 d'aquests camins.

Doneu un algorisme tan eficient com pugueu que, donats G, E', B, C i D, determini si és possible obtenir els camins requerits i que, en cas que ho sigui, retorni, per a cada $c \in C$, el camí que comença a c i acaba a alguna cruïlla de B.