Algorísmia QT 2018–2019

Examen final **8 de Gener de 2019**

Durada: 2h 50m

Instruccions generals:

- L'exercici 4 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 (Ex1, Ex2, Ex3 i Ex4).
- La puntuació total d'aquest examen és de **10 punts**.

Exercici 1 (2 punts) (Revisió) Un professor rep n sol·licituds de revisions d'examen. Abans de començar, el professor mira la llista dels n estudiants que han sol·licitat revisió i pot calcular, per a cada estudiant i, el temps t_i que utilitzarà per atendre l'i-èsim estudiant. Per a estudiant i, el temps d'espera e_i és el temps que el professor triga a revisar els exàmens dels estudiants que fan la revisió abans que i.

Dissenyeu un algorisme per a computar l'ordre en que s'han de revisar els exàmens dels n estudiants de manera que es minimitzi el temps total d'espera: $T = \sum_{i=1}^{n} e_i$.

Exercici 2 (2 punts) (Matrioshka) En Dilworth és el col·leccionista més destacat del món de matrioshkas, les nines russes nidificades, com les de la figura de sota.



En té milers de nines buides de fusta de diferents mides. Per construir un matrioshka la nina més petita es fica dintre de la segona més petita, i aquesta nina, al seu torn, es fica dintre de la la següent i així successivament.

En Dilworth es pregunta si hi ha una altra manera de nidificar-les perquè acabi amb el màxim possible de nines nidificades. Després de tot, això faria que pogués emmagatzemar millor la seva col·lecció i ampliar-la per tal que fos encara més magnífic!

Per a cada nina tenim mesures de la seva amplada i la seva altura. Una nina amb amplada w_i i altura h_i encaixa en una altra nina d'amplada w_j i alçada h_j si i només si $w_i < w_j$ i $h_i < h_j$. Donades les mides de les nines proporcioneu un algorisme, tan eficient com pugueu, per construir la matrioshka amb el màxim nombre possible de nines nidificades.

Exercici 3 (2 punts) (MenjaBe) MenjaBe produeix una gran varietat de menús de menjars diferents. Malauradament, només poden produir els seus menús en quantitats limitades, de manera que solen quedar-se sense els més populars, deixant clients insatisfets. Per minimitzar aquest problema, MenjaBe vol implementar un sofisticat sistema de distribució de dinars. Els clients haurien d'enviar un missatge de text amb les seves opcions de menús acceptables abans de l'hora de dinar. A continuació, fan una assignació de dinars als clients. MenjaBe té decidit compensar amb un val de 5 euros als clients als qui no pot assignar cap de les seves opcions. MenjaBe vol minimitzar la quantitat de vals que dóna.

Doneu un algorisme eficient per a assignar menús als clients en un dia. En general, en un dia determinat, MenjaBe sap que ha produït m tipus de menús i la quantitat de cadascun d'ells $q_1, \dots, q_m$. A més els n clients envien un text amb les seves preferències, el client i indica un conjunt A_i de menús acceptables. L'algorisme ha d'assignar a cada client una de les seves opcions o un val de 5 euros, de manera que es minimitzi el nombre de vals.

Cognoms, Nom Data

Exercici 4 (4 punts)

(a) (0.5 punts) L'ordenació RADIX funciona correctament quan, per a ordenar els dígits, utilitzem qualsevol algorisme d'ordenació que sigui correcte.

(b) (0.5 punts) Donat un vector d' enters $A[1,\ldots,n]$, la complexitat d'ordenar-lo utilitzant comptatge és $O(n)$.

(c) (0.5 punts) Suposem que tenim n intervals $I_i = (e_i, d_i)$, $1 \leq i \leq n$, i que cada interval té associat un pes w_i. Volem trobar un conjunt S d'intervals que no es sobreposen i que maximitzi $\sum_{i \in S} w_i$. Considereu el següent algorisme: Comencem amb el conjunt T que conté tots els intervals. A cada pas, seleccionem de T un interval I_j amb màxim w_j i ho afegim a S. Després eliminem de T l'interval I_j i tots els intervals a T que es sobreposin amb I_j. Repetim fins a no poder seleccionar-ne més. Quin tipus d'esquema algorísmic és aquest algorisme? Es cert que aquest algorisme produeix una solució òptima del problema?

(d) (0.5 punts) Si utilitzem l'algorisme de Huffmann per a comprimir un text format per n símbols que apareixen amb freqüències $f_1, f_2, f_3, \ldots, f_n$ respectivament, quina és la màxima longitud de compressió d'un símbol que podem obtenir? Doneu un algorisme per obtenir freqüències on es compleixi aquesta condició per a un valor de n donat.

(e) (0.5 punts) Sigui $G = (V, E)$ un graf amb pesos $w(E)$ i sigui T un arbre d'expansió amb cost mínim (MST) a G. Aleshores el camí de pes mínim a G entre dos vèrtexs v_1 i v_2 ha de ser també un camí mínim a T.

(f) (0.5 punts) Donat un graf no dirigit $G = (V, E)$, definim el seu *diàmetre* d com la màxima distància entre qualsevol parell de vèrtexs de G. Doneu un algorisme, el més eficient possible, que amb entrada un graf no dirigit qualsevol G amb n vèrtexs i m arestes, calculi el diàmetre de G.

(g) (0.5 punts) És cert que si a una xarxa de flux tots els arcs tenen capacitats amb valors diferents, el flux amb valor màxim és únic?

(h) (0.5 punts) Què és un algorisme d'aproximació? Explica per què l'estratègia voraç, pot ser bona per a produir algorismes d'aproximació per a problemes que són NP-complets. Recordes algun exemple vist a classe?