Cognoms, Nom DNI

Algorísmia QT 2016–2017

Examen final **9 de Gener de 2017**

Durada: 2h 50m

Instruccions generals:

- Els exercicis 1 i 2 s'han 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 ès una variació haureu de donar els detalls.
- Es valorarà especialment la claredat i concisió de la presentació.
- Entregueu per separat la solució de cadascun dels exercicis.

Exercici 1 (3.5 punts)

(a) (0.5 punts) Siguin A i B dos conjunts d'enters. Doneu un algorisme que, amb temps esperat $O(n)$, ens digui si els dos conjunts contenen els mateixos elements.

Digueu i justifiqueu si cadascuna de les afirmacions següents són certes o falses.

(b) (0.5 punts) Donat un graf $G = (V, A)$ amb $w : A \to \mathbb{Z}^+$ i un conjunt $S \subseteq V$, si $e = (u, v)$ és una aresta tal que e té el cost mínim entre S i $V - S$, digueu si l'arbre d'expansió miním (Minimum Spanning Tree) a G conté l'aresta e.

(c) (0.5 punts) Tenim una xarxa $(G = (V, E), s, t, \{c(e)\})$ amb $\forall e \in E$, $c(e) \in \mathbb{Z}^+$. Sigui (A, B) un tall mínim $s - t$. Si afegim 1 a totes les capacitats, aleshores (A, B) encara és un tall mínim $s - t$ respecte a les noves capacitats $c(e) + 1$.

(d) (0.5 punts) Donat un graf dirigit $\vec{G} = (V, \vec{E})$, amb pesos a les arestes $w(e) \in \mathbb{R}$, la manera més ràpida de trobar les distàncies mínimes entre tots els parells de vèrtexs és aplicar n cops l'algorisme de Dijkstra.

Cognoms, Nom DNI

Doneu una explicació breu i concisa

(e) (0.5 punts) Tenim n caixes ($c_1, \ldots, c_n$), una d'elles conté un bitllet de 500 euros i les altres estan buides. Volem trobar el bitllet obrint el mínim nombre de caixes. Considereu l'algorisme següent:

- Seleccionem $x \in \{0, 1\}$ amb distribució uniforme.
  - Si $x = 0$ obrim caixes en l'ordre $\{c_1, \ldots, c_n\}$ fins trobar el bitllet.
  - Si $x = 1$ obrim caixes en l'ordre $\{c_n, \ldots, c_1\}$ fins trobar el bitllet.

Quin es el nombre esperat de caixes que s'obren en aquest algorisme?

(f) (0.5 punts) Que és un *blockchain*?

(g) (0.5 punts) Donat un graf no-dirigit amb dos vèrtexs distingits s i t, explica com podríeu utilitzar teoria de fluxos per a trobar tots els camins disjunts entre s i t.

Exercici 2 (1 punt) La suma **o-exclusiva** $\oplus$ és un operador lògic on $0 \oplus 1 = 1 \oplus 0 = 1$ i $0 \oplus 0 = 1 \oplus 1 = 0$, per tant donades cadenes de bits B_1 i B_2, $B_1 \oplus B_2$ consisteix en l'aplicació bit a bit de l'operador $\oplus$ (per exemple $10110 \oplus 00111 = 10001$). Donat un missatge M (com a cadena de bits) definim la funció h_c de la manera següent:

- Fem una partició de M en k blocs $M = B_1||B_2||\cdots||B_k$, (on $||$ és l'operador concatenació) i on cada bloc B_i conté exactament de 5 bits. Si la grandària de M no és múltiple de 5, afegim 0s la dreta, fins a assolir la grandària desitjada.
- $h_c(M) = B_1 \oplus B_2 \oplus \cdots \oplus B_k$.

(a) (0.25 punts) Quina és la sortida de $h_c(10101110)$?.

(b) (0.75 punts) És h_c una bona funció criptogràfica? (és a dir, té les propietats següents: (1) Per a una entrada M amb grandària qualsevol, sempre dóna una sortida amb la mateixa grandària; (2) $h_c(M)$ es pot calcular fàcilment; (3); h_c^{-1} és difícil de calcular; (4) coneixent h_c i M, és difícil calcular una M' tal que $h_c(M') = h_c(M)$; (5) si canviem un bit a M aleshores $h_c(M)$ canvia molt.)

Exercici 3 (2,5 punts)

Tenim un programa que permet la simulació d'un sistema físic de temps discret. Volem simular tants passos del sistema com sigui possible. El nostre laboratori te accés a dos supercomputadors (A i B) capaços de processar el treball. No obstant això, són màquines compartides i no sempre poden executar els nostres treballs amb la prioritat més alta. Tant A com B poden processar el nostre programa.

Suposem que sabem, pels següents n minuts, la potència de processament disponible a cada màquina. Al minut i, podem executar a_i passos de la simulació a A o bé b_i a B. La simulació es pot transferir d'una màquina a un altra però, per fer-ho, s'ha de salvar i restaurar l'estat i això té un cost d'un minut de temps en el què no es pot fer cap progrés en la simulació. Volem un pla d'execució pels n minuts següents. Aquest pla ha de indicar, per a cada minut, A o B o "mou", i ha de ser consistent amb les restriccions donades. A més, volem que maximitzi el nombre total de passos de simulació executats.

(a) (0,5 punts) Demostreu que el següent algorisme no resol correctament el problema proposat.

```
1:  procedure PLA D'EXEC(a, b)
2:      if a[1] ≥ b[1] then
3:          s[1] = 'A'
4:      else
5:          s[1] = 'B'
6:      end if
7:      i = 2
8:      while i ≤ n do
9:          if s[i − 1] == 'A' then
10:             if b[i + 1] > a[i] + a[i + 1] then
11:                 s[i] = 'mou'; s[i + 1] = 'B'; i = i + 2
12:             else
13:                 s[i] = 'A'; i = i + 1
14:             end if
15:         else
16:             Com al cas previ canviant A/a per B/b
17:         end if
18:     end while
19: end procedure
```

(b) (2 punts) Doneu un algorisme eficient que, donats $a_1, \ldots, a_n$ i $b_1, \ldots, b_n$, proporcioni un pla d'execució que permeti executar el màxim nombre de passos de simulació.

Exercici 4 (3 punts) SuperFast és una empresa de transport que està intentant decidir si li interessa participar en una oferta de treball de Amazon a Barcelona. Amazon voldria subcontractar el transport de productes de proximitat a SuperFast. El transport ha de garantir uns compromisos de puntualitat molt estrictes i SuperFast vol una estimació de la quantitat de nous vehicles que hauria d'incorporar a la seva flota i una estimació del cost corresponent al seu ús. Amazon li ha proporcionat els resultats de les seves simulacions de comportament dels clients i, en particular, de les necessitats de transport diàries per uns quants dies. Per un dia es disposa d'una llista de sol·licituds de transport. Cada sol·licitud de transport especifica les coordenades GPS de l'origen i del destí d'un enviament juntament amb l'hora de recollida i d'entrega. SuperFast disposa d'un programa que li proporciona el temps necessari de desplaçament d'un vehicle entre qualsevol parell de posicions de la ciutat coneixent el temps d'inici del trasllat.

En aquest primer estudi SuperFast assumeix el cas pitjor en el què els vehicles no poden portar més d'un enviament. A més, el vehicle ha de ser a la posició d'origen al temps estipulat i no pot deixar el punt de destí fins el temps estipulat de trasllat. Així, un vehicle que transporti un enviament pot encarregar-se d'un altre sempre que el temps de desplaçament entre el destí i el nou origen li permeti arribar l'hora estipulada. També assumeix que l'estimació del temps necessari de desplaçament és acurada.

Disenyeu algorismes amb cost polinòmic que:

(a) (1,5 punts) Donats un nombre k i les sol·licituds de transport d'un dia, determini si es poden servir amb k vehicles.

(b) (0,5 punt) Donades les sol·licituds de transports d'un dia, determini el nombre mínim de vehicles que les poden servir ($k_{\min}$)

(c) (1 punt) Donades les sol·licituds de transports de un dia, proporcioni una planificació per a cadascun dels $k_{\min}$ vehicles indicant, per a cada vehicle, la seqüència de sol·licituds de transport que ha de servir i el temps total de desplaçament del vehicle.

Podeu suposar que el càlcul del temps de desplaçament entre dos posicions té un temps constant i que entre l'hora de recollida i la d'entrega hi ha temps suficient per fer-hi el desplaçament amb puntualitat.

Ajut: Podeu pensar un vehicle com una unitat de flux i una sol·licitud de transport com un arc amb fites inferiors i superiors a la seva capacitat.