

# Algorísmia Q1 2020–2021

**Examen Parcial**

**4 de novembre de 2020**

Durada: 1:30

---

Instruccions generals:

- 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.
- 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 (3.5 punts).** Ja sabeu que fer la fusió ordenada de dues seqüències ordenades d' $m$  i  $n$  elements, respectivament, comporta fer  $m + n$  moviments de dades (penseu, per exemple, en un *merge* durant l'ordenació amb *mergesort* d'un vector). Però si hem de fer la fusió d' $N$  seqüències, dos a dos, l'ordre en què es facin les fusions és rellevant. Imagineu que tenim tres seqüències  $A$ ,  $B$  i  $C$  amb 30, 50 i 10 elements, respectivament. Si fusionem  $A$  amb  $B$  i després el resultat el fusionem amb  $C$ , farem  $30 + 50 = 80$  moviments per a la primera fusió i  $80 + 10 = 90$  per a la segona, amb un total de 170 moviments. En canvi, si fusionem primer  $A$  i  $C$  i el resultat el fusionem amb  $B$  farem un total de 130 moviments.

Dissenyau un algorisme golafre (*greedy*) per fer les fusions i obtenir la seqüència final ordenada amb mínim nombre total de moviments. Justifiqueu la seva correctesa i calculeu-ne el cost temporal del vostre algorisme en funció del nombre de seqüències  $N$ .

**Exercici 2 (2 punts).** Hi ha un concurs de TV amb  $n$  participants on cada participant escull un enter entre 0 i 1000000. El premi és per als dos concursants que escullen els enters més propers. Dissenyau un algorisme que, en temps lineal, li digui al presentador quins són els dos concursants guanyadors.

**Exercici 3 (2 punts)** Supposeu que comencem un procés dinàmic per formar un graf  $G$  a partir d'un conjunt  $V$  de  $n$  vèrtexs i una seqüència d'arestes  $S$  donats. A cada pas del procés se'ns proporciona una nova aresta del graf, fins a introduir les  $m$  arestes de la seqüència  $S = \{e_1, e_2, \dots, e_m\}$ . Així doncs, al llarg del procés tenim una seqüència de grafs  $G_0, G_1, \dots, G_m$ , on  $G_0 = (V, \emptyset)$  i al pas  $t$ , obtenim  $G_t$  afegint l'aresta  $e_t$  a  $G_{t-1}$ .

Doneu un algorisme, tan eficient com pugueu, per a poder obtenir el nombre de components connexes a cada pas de procés.

**Exercici 4 (2.5 punts).** Tenim un vector  $A = (a_1, \dots, a_n)$  d'elements d'un conjunt sobre els quals s'ha definit una relació d'ordre, i un vector  $R = (r_1, \dots, r_p)$  d'enters i ordenat, amb  $1 \leq r_1 < r_2 < \dots < r_p \leq n$ .

Proporcioneu un algorisme que, donats  $A$  i  $R$ , i amb cost  $o(pn)$ , trobi l' $r_1$ -èsim,  $r_2$ -èsim,  $\dots$ ,  $r_p$ -èsim del vector  $A$ . Justifiqueu la correctesa de l'algorisme proposat i el seu cost en funció de  $n$  i de  $p$ .