

Cognoms

Nom

DNI

Examen Final EDA

Duració: 3h

12/06/2023

-
- L'enunciat té 4 fulls, 8 cares, i 4 problemes.
 - Poseu el vostre nom complet i número de DNI a cada full.
 - Contesteu tots els problemes en el propi full de l'enunciat a l'espai reservat.
 - Llevat que es digui el contrari, sempre que parlem de cost ens referim a cost asimptòtic en temps.
 - Llevat que es digui el contrari, **cal justificar les respostes.**
-

Problema 1

(2 pts.)

Responen les preguntes següents utilitzant, quan calgui, els teoremes mestre adjunts:

(a) (1 pt.) Considereu la funció següent:

```
int f(const vector<int>& v, int e, int d) {  
    if (d ≤ e) return 1;  
    return f(v,  $\overline{A}$ ,  $\overline{B}$ ) + f(v,  $\overline{C}$ ,  $\overline{D}$ );  
}
```

Ompliu les caixes A, B, C, D per tal que, donat un vector v de mida n , una crida $f(v, 0, v.size() - 1)$ tingui cost $\Theta(\log n)$. Feu el mateix per a cost $\Theta(n)$.

(b) (1 pt.) Considereu el codi següent:

```
bool cerca2 (int x, const vector<int>& v, int e, int d) {  
    for (int i = e; i ≤ d; ++i)  
        if (v[i] == x) return true;  
    return false; }
```

```
bool cerca3 (int x, const vector<int>& v, int e, int d) {  
    if (e > d) return false;  
    int m = (e+d)/2;  
    if (v[m] == x) return true;  
    else if (v[m] < x) return cerca3(x, v, m+1, d);  
    else return cerca3(x, v, e, m-1); }
```

```
bool cerca (int x, const vector<int>& v, int e, int d) {  
    if (d - e < 2) {  
        for (int i = e; i ≤ d; ++i)  
            if (v[i] == x) return true;  
        return false;  
    }  
    int n = d - e + 1, p1 = e + n/3, p2 = d - n/3;  
    if (cerca2(x, v, e, p1 - 1)) return true;  
    if (cerca3(x, v, p1, p2)) return true;  
    return cerca(x, v, p2 + 1, d); }
```

Si v és un vector de mida n , quin és el cost en cas pitjor, en funció d' n , d'una crida a $cerca(x, v, 0, v.size() - 1)$?

Cognoms

Nom

DNI

Problema 2

(2 pts.)

Considerem una implementació d'arbres binaris de cerca on el nodes tenen l'estructura següent:

```
struct Node {  
    int key;  
    Node* left ; // Punter al fill esquerre  
    Node* right ; // Punter al fill dret  
};
```

Us demanem que, a partir d'un arbre binari de cerca amb n nodes, construïu un *max-heap* que contingui totes les claus de l'arbre en temps $\Theta(n)$. Si hi ha més d'un *max-heap* possible, escolliu el que vulgueu. Recordeu que un *heap* es pot implementar com un vector. Heu d'implementar una funció

```
vector<int> to_Heap (Node* n);
```

on n és un punter a l'arrel de l'arbre binari de cerca. Podeu utilitzar funcions auxiliars. Us demanem codi en C++. Descripcions a alt nivell rebran 0 punts.

Aquesta cara estaria en blanc intencionadament si no fos per aquesta nota.

Cognoms

Nom

DNI

Problema 3

(3 pts.)

En aquest problema representarem un graf dirigit amb nodes $\{0, 1, \dots, n-1\}$ com una matriu d'adjacència $n \times n$, on la fila i , columna j , indica si existeix un arc $i \rightarrow j$.

typedef *vector*<*vector*<**bool**>> *Graf*;

Un graf *torneig* és un graf dirigit on per a cada parell de vèrtexs diferents hi ha exactament un arc, i sense arcs des de cap vèrtex cap a ell mateix.

- (a) (1 pt.) Escriviu una funció que determini si un graf donat d' n vèrtexs és un graf torneig en temps $\Theta(n^2)$ en el cas pitjor.

bool *es_torneig* (**const** *Graf*& *G*) {

}

- (b) (1 pt.) Demostreu per inducció en el nombre de vèrtexs que tot graf torneig té un camí Hamiltonià, és a dir, un camí que visita tots els vèrtexs exactament una vegada (pista: mostreu que un nou vèrtex es pot inserir en un camí d' $n-1$ vèrtexs per obtenir un camí d' n vèrtexs).

- (c) (1 pt.) Valent-vos de la demostració anterior, doneu un algorisme de cost com a molt $\Theta(n^2)$ que retorni un camí Hamiltonià d'un graf torneig. Us pot ser útil representar el camí com una llista de nodes (enters). No és necessari que doneu codi, una descripció a alt nivell serà suficient. Justifiqueu el cost, en cas pitjor, del vostre algorisme.

Cognoms

Nom

DNI

Problema 4

(3 pts.)

Per a cadascuna de les preguntes següents, responeu raonadament si són certes, falses o no ho sabem. En cas de ser certes, indiqueu dos possibles problemes A i B que compleixin la propietat mencionada. En cas de no saber-se, expliqueu què implicaria que fos certa i què implicaria que fos falsa.

a) (1 pt.) **Existeixen** dos problemes diferents A i B tals que:

- $A \in P$
- $B \in NP$ -difícil
- B es pot reduir polinòmicament cap a A

b) (1 pt.) **Existeixen** dos problemes diferents A i B tals que:

- $A \in P$
- $B \in NP$ -complet
- A es pot reduir polinòmicament cap a B

c) (1 pt.) **Existeixen** dos problemes diferents A i B tals que:

- $A \in NP\text{-complet}$
- $B \in NP\text{-difícil}$
- A es pot reduir polinòmicament cap a B i B cap a A .