

Cognoms

Nom

DNI

Examen Parcial EDA

Duració: 1h 30min

03/11/2022

-
- L'enunciat té 3 fulls, 6 cares, i 2 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

(4 pts.)

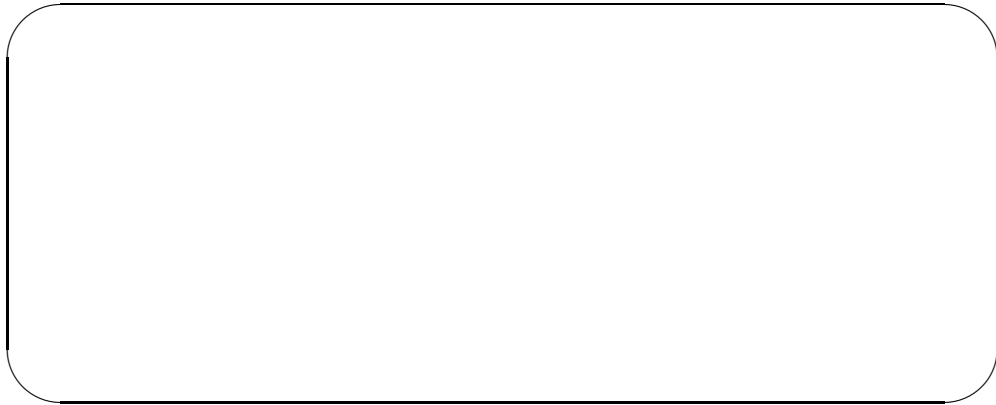
Responen les preguntes següents:

(a) (2 pts.) Considereu el codi següent:

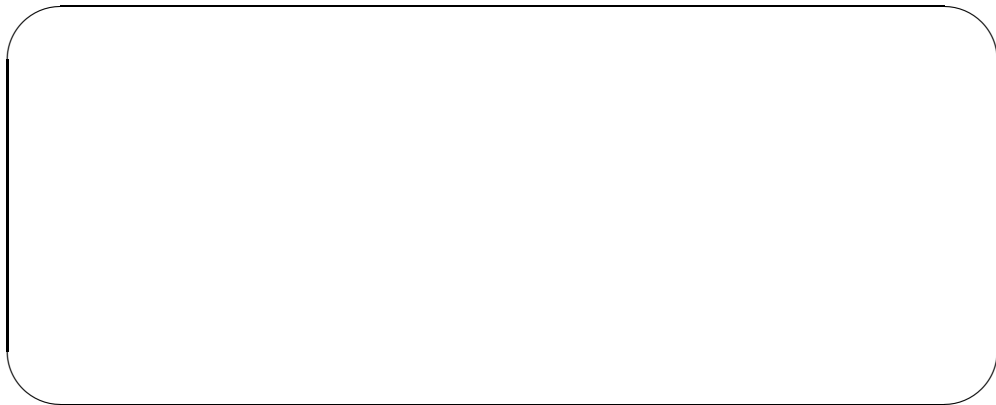
```
int f (int x, int n) {  
    if (n == 1) return x;  
    else {  
        int tmp = f(x, n-1);  
        int res = 0;  
        for (int i = 0; i < x; ++i) res += tmp;  
        return res;  
    }  
}  
  
int main(){  
    int N;  
    cin >> N;  
    cout << f(N, N) << endl;  
}
```

Si assumim que $N \geq 1$, què escriu per pantalla el programa anterior? Justifica la teva resposta.

Quin és el cost del programa anterior en funció d' N ?



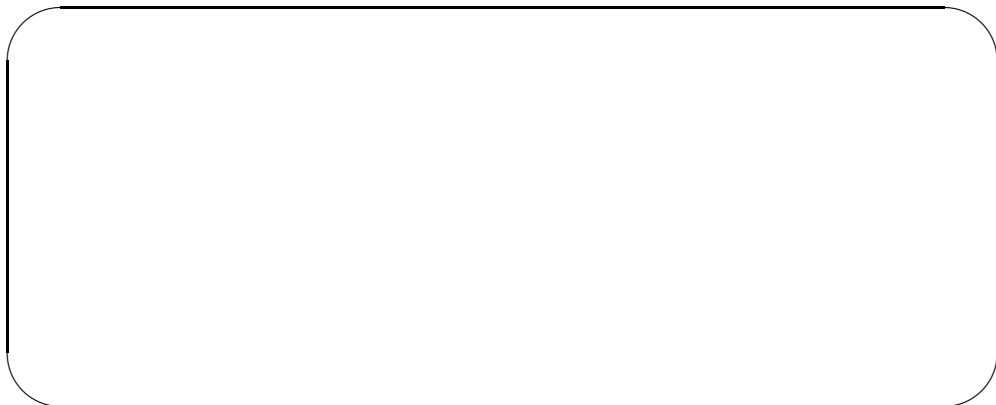
(b) (2 pts.) Sigui $f(n) = \ln(\ln(n^2))$ i $g(n) = \ln(\ln n)$. Podem afirmar que $f(n) \in \Theta(g(n))$?



Definim ara

$$F(n) = \begin{cases} n^2, & \text{si } 0 \leq n \leq 10 \\ f(n) & \text{si } n > 10 \end{cases} \quad G(n) = \begin{cases} n^3, & \text{si } 0 \leq n \leq 10 \\ g(n) & \text{si } n > 10 \end{cases}$$

on f i g són les funcions anteriorment definides. Podem afirmar que $F(n) \in \Theta(G(n))$?



Cognoms

Nom

DNI

Problema 2

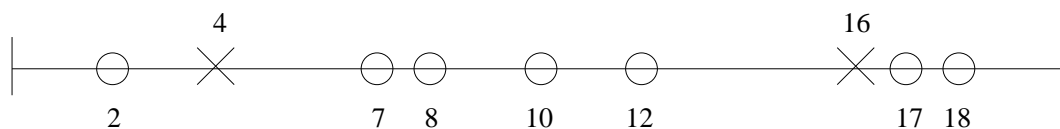
(6 pts.)

L'ajuntament d'una gran ciutat decideix comprar estufes de carrer per tal d'escalfar totes les escoles del municipi. Totes les escoles estan situades sobre una mateixa recta, i disposem d'un vector s d'enters amb les distàncies de totes elles al punt quilomètric zero. Sabem que els elements d' s són tots diferents.

Els assessors en matèria energètica han decidit ja quantes estufes comprar i en quin punt de la recta situar-les. Disposem d'un vector h que indica el punt quilomètric en la recta de cadascuna de les estufes. Altra vegada, tots els elements d' h són diferents.

Per estalviar energia, podem ajustar les estufes pes tal d'escalfar qualsevol escola que estigui a distància com a molt d . Si volem ajustar totes les estufes de la mateixa manera, quina és la mínima d que ens permet escalfar totes les escoles?

Gràficament, si $s = [8, 17, 2, 12, 18, 7, 10]$ i $h = [16, 4]$ tenim la situació



i podem veure que la solució és $d = 6$. Si prenem $d = 5$, per exemple, l'escola a la posició 10 no seria escalfada per cap estufa.

Per simplificar els raonaments, assumiren en tot aquest problema que $n = s.size() = h.size()$.

(a) (0.5 pts.) La següent funció ens proporciona una solució senzilla al problema:

```
int radius (const vector<int>& h, const vector<int>& s) {
    int rad = 0;
    for (int i = 0; i < s.size (); ++i) {
        int rad_s = inf; // inf és l'int més gran
        for (int j = 0; j < h.size (); ++j)
            rad_s = min(rad_s, abs(h[j] - s[i]));
        rad = max(rad, rad_s);
    }
    return rad;
}
```

En funció d' n , quin és el cost d'una crida a aquesta funció?

- (b) (2 pts.) Assumim en aquest apartat que tant h com s estan ordenats de manera creixent. Ompliu els buits de la funció següent perquè resolgui el problema que tenim entre mans:

```

int radius_2 (const vector<int>& h, const vector<int>& s) {
    int r = 0, j = 0;
    for (int i = 0; i < s.size (); ++i){
        while (j < h.size () and h[j] < s[i]) ++j;
        int rad;
        if (j == h.size ()) rad = 
        else if (j == 0) rad = 
        else rad = 
        r = max(r,rad);
    }
    return r;
}

```

Quin és el cost en cas pitjor d'una crida a *radius_2* en funció d' n ?

- (c) (3 pts.) Assumim en aquest apartat que h està ordenat de forma creixent. Donada una escola a la posició p , volem trobar la mínima k tal que $p \leq h[k]$ i $0 \leq k < n$. Si no existeix cap k que compleixi aquestes dues condicions (és a dir, si p és més gran que qualsevol element d' h) cal retornar n .

Ompliu els buits de la funció següent perquè retorni aquest valor k en temps $\Theta(\log n)$ en cas pitjor. Solucions que no tinguin aquest cost rebran zero punts.

```

int find (const vector<int>& h, int p) {return find(h, 0, h.size ()-1, p);}

```

Cognoms

Nom

DNI

```
int find(const vector<int>& h, int l, int r, int p) {  
    if (r < l) {
```

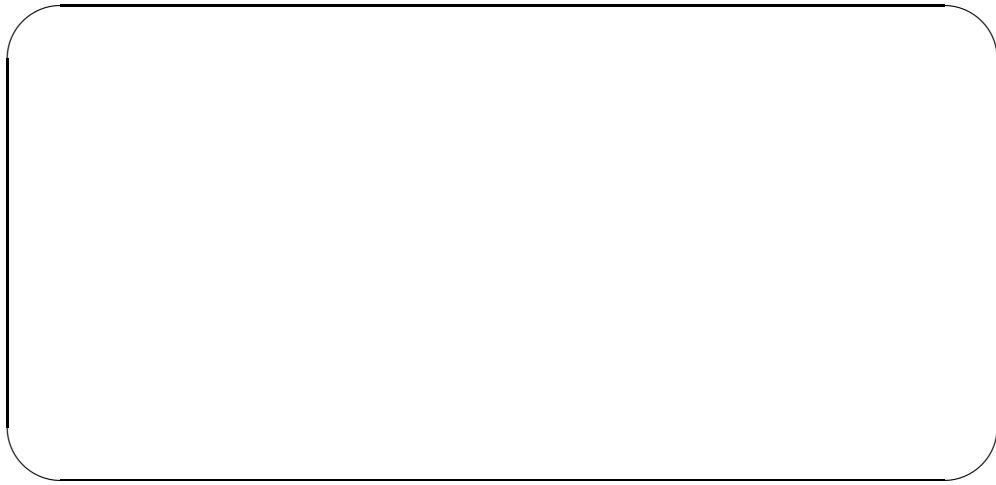
```
    }
```

```
    else {
```

```
    }
```

```
}
```

Justifica per què la teva solució té cost, en cas pitjor, $\Theta(\log n)$.



- (d) (0.5 pts.) Si en la funció *radius_2* de l'apartat (b) reemplacem la línia del **while** per $j = \text{find}(h, s[i])$; quin seria el cost en cas pitjor d'una crida a *radius_2* en funció d' n ?

