

Cognoms

Nom

DNI

Examen Parcial EDA

Duració: 1h 30min

08/11/2021

-
- L'enunciat té 4 fulls, 8 cares, i 2 problemes.
 - Poseu el vostre nom complet i número de DNI a cada problema.
 - 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

(5 pts.)

Responen les preguntes següents:

(a) (1 pt.) Considereu la funció *mystery*:

```
bool mystery (int n) {  
    if (n ≤ 1) return false;  
    if (n == 2) return true;  
    if (n%2 == 0) return false;  
    for (int i = 3; i*i ≤ n; i += 2)  
        if (n%i == 0) return false;  
    return true;  
}
```

La funció *mystery* determina si i el cost en el cas pitjor, en funció d' n , és $\Theta(\text{ })$. Justifiqueu aquest cost:

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

```
int j = 0;
int s = 0;
for (int i = 0; i < n; ++i)
    if (i == j*j) {
        for (int k = 0; k < n; ++k) ++s;
        ++j;
    }
```

En funció d' n , el cost és $\Theta(\text{ } \boxed{\text{ }} \text{ })$. Justifiqueu la vostra resposta:

(c) (1 pts.) Donat un vector v d' n enters, volem calcular el nombre total de parelles (i, j) tals que $0 \leq i < j \leq n - 1$ i $v[i] = v[j]$. Per tal de solucionar el problema ens donen el codi següent:

```
int pairs (const vector<int>& v, int l, int r) {
    if (l >= r) return 0;
    else {
        int m = (l+r)/2;
        int n_left = pairs(v, l, m);
        int n_right = pairs(v, m+1, r);
        int n_crossed = 0;
        for (int i = l; i <= m; ++i)
            for (int j = m+1; j <= r; ++j)
                if (v[i] == v[j]) ++n_crossed;
        return n_left + n_right + n_crossed;
    }
}

int pairs (const vector<int>& v) {return pairs(v, 0, v.size() - 1);}
```

Cognoms

Nom

DNI

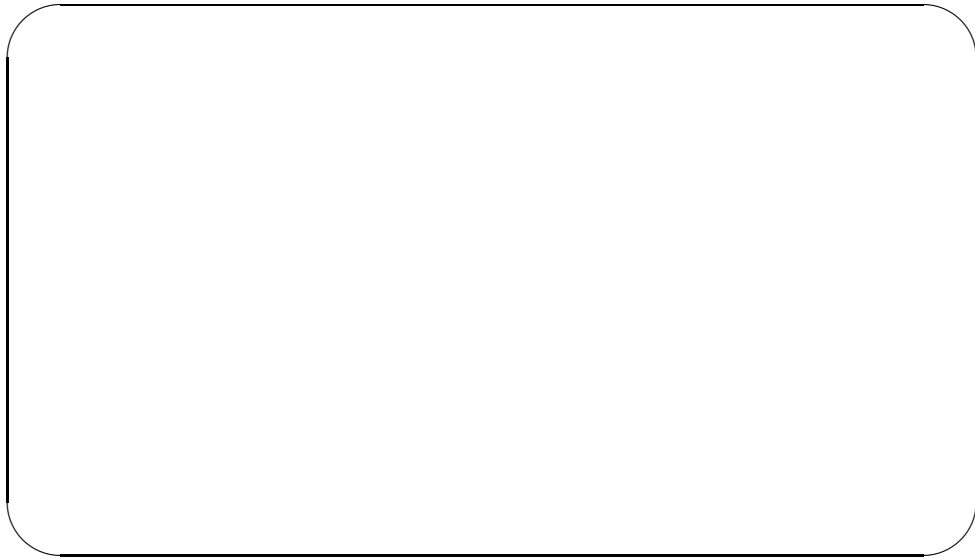
--	--	--

En funció d' n , el cost d'una crida a $\text{pairs}(v)$ és $\Theta(\quad)$. Justifiqueu la vostra resposta:

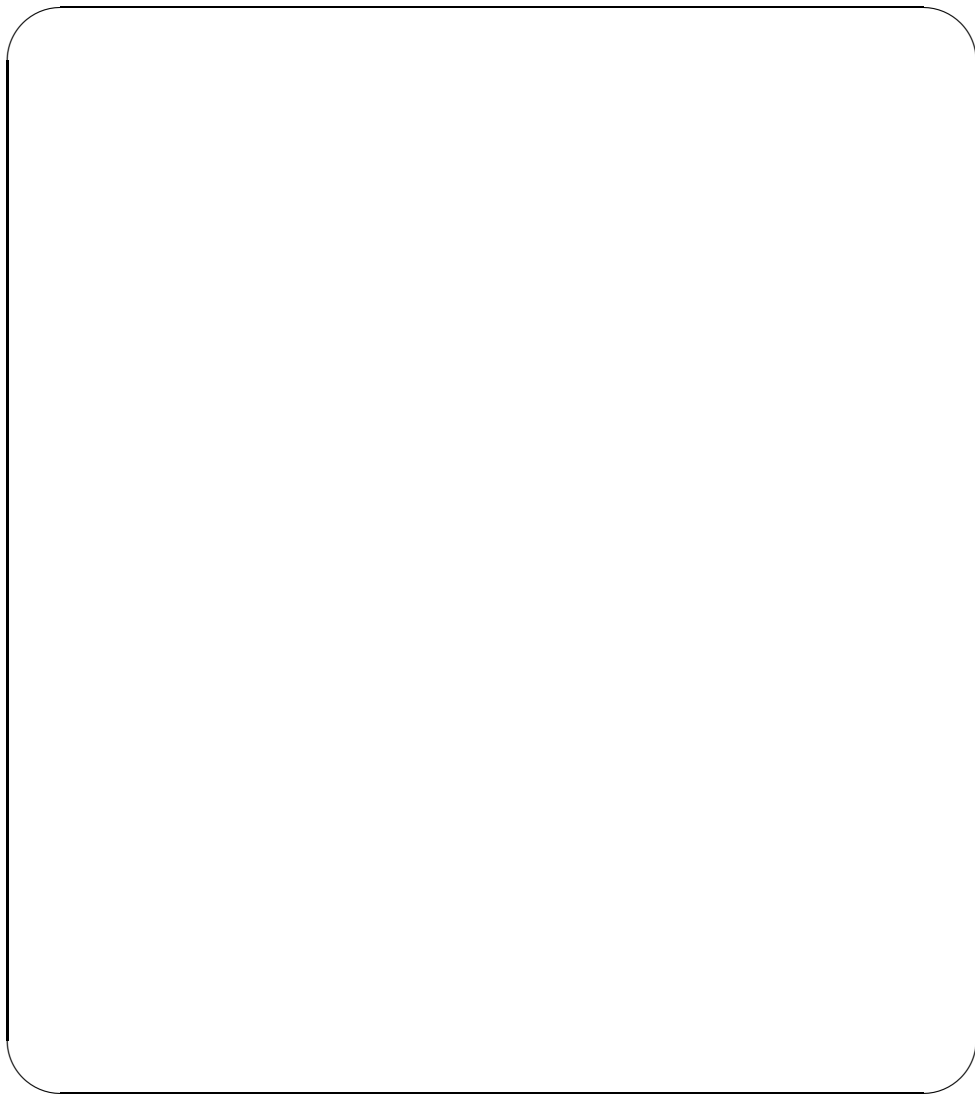
- (d) (2 pts.) Ens asseguruen ara que tots els nombres de v són naturals estrictament menors que un cert paràmetre enter K , que és constant i no depèn d' n . En aquesta situació, el codi següent és una solució al problema de l'apartat anterior:

```
int pairs_2 (const vector<int>& v) {  
    vector<int> times(K,0);  
    int n = v.size ();  
    for (int i = 0; i < n; ++i) ++times[v[i ]];  
  
    int res = 0;  
    for (int i = 0; i < K; ++i) res += (times[i ]*(times[i ]-1))/2;  
    return res ;  
}
```

Si n és la mida de v , el cost de pairs_2 en funció d' n és $\Theta(\quad)$. Justifiqueu la vostra resposta:



Expliqueu per què el codi anterior és una solució correcta:



Cognoms**Nom****DNI****Problema 2****(5 pts.)**

Donat un vector v d' n enters ordenats creixentment i un enter x , volem determinar el nombre de vegades que x apareix a v .

- (a) (1.5 pts.) Diem que la *primera aparició* d' x dins v és el menor índex i amb $0 \leq i < n$ i $v[i] = x$, o bé -1 si x no apareix a v . Un amic ens comenta que si sabem trobar la primera aparició, aleshores trobar una solució eficient al nostre problema no és massa difícil. Ompliu el codi següent per tal que trobi la primera aparició d' x dins v de manera que el seu cost en cas pitjor sigui $O(\log n)$.

```
int first_occurrence (int x, const vector<int>& v, int l, int r) {
    if (l > r) return -1;
    else {
        int m = (l+r)/2;
        if (v[m] < x) return first_occurrence (x,v,m+1,r);
        else if (v[m] > x) return first_occurrence (x,v,l,m-1);
    }
}
```

```
int first_occurrence (int x, const vector<int>& v) {
    return first_occurrence (x,v,0,v.size()-1);
}
```

- (b) (1.5 pts.) Amb la funció anterior funcionant ja correctament, ens demanen que omplim el codi següent per tal que calculi el nombre d'aparicions d' x dins v :

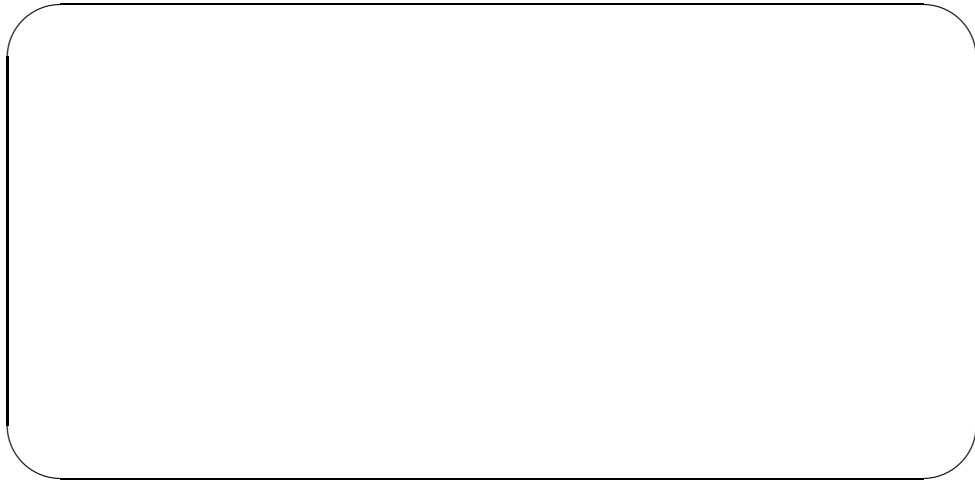
```
int p = first_occurrence (x,v);

int n = v.size ();
for (int i = 0; i < n; ++i) v[i] = -v[i];
for (int i = 0; i <= n/2 - 1; ++i) swap(v[i], v[n-1-i]);
int q = first_occurrence (-x,v);

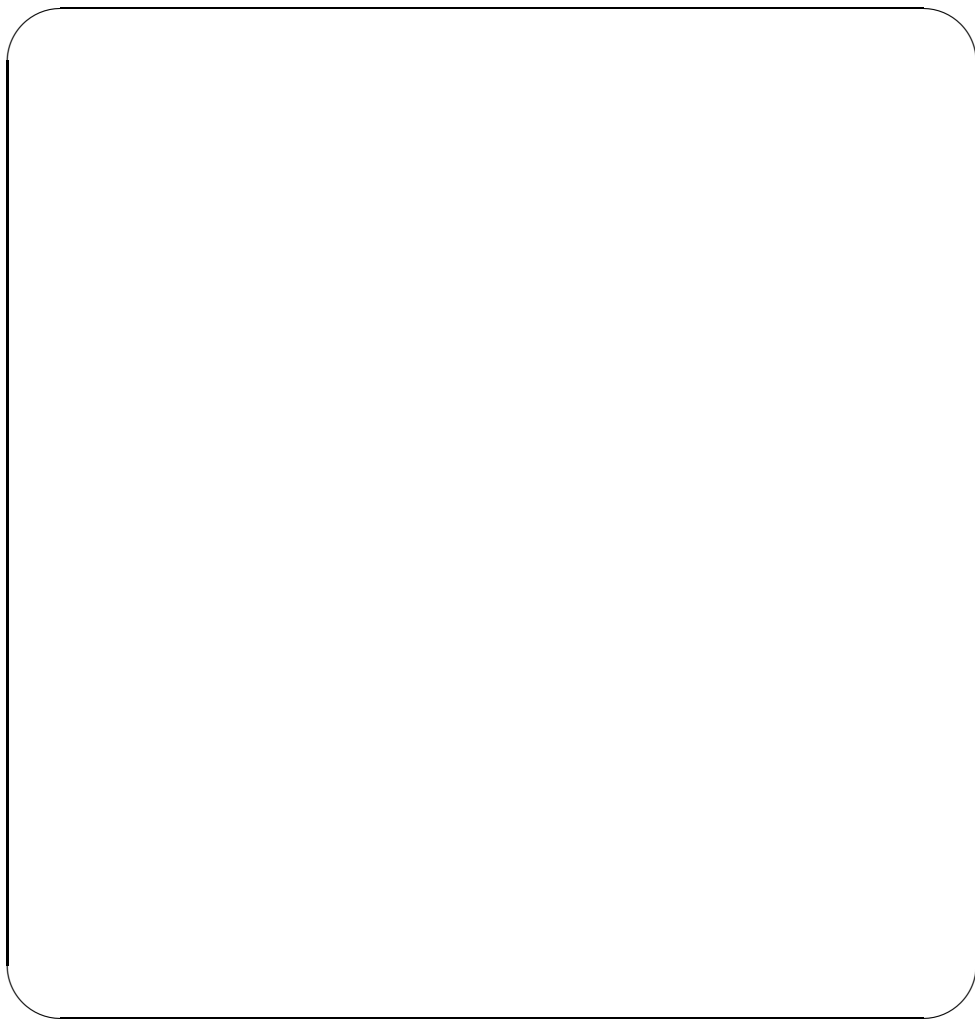
int res;
if (p == -1) res = 0;
else res = 

cout << res << endl;
```

Determineu de forma raonada, el cost en cas pitjor del codi anterior en funció d' n .

A large, empty rounded rectangular box with a thin black border, intended for the student to provide a reasoned answer to the question above.

Expliqueu per què el codi anterior calcula correctament el nombre d'aparicions d' x dins v .

A large, empty rounded rectangular box with a thin black border, intended for the student to explain why the previous code correctly calculates the number of occurrences of x in v .

Cognoms

Nom

DNI

- (c) (2 pts.) Existeix un algorisme per calcular el nombre d'aparicions que sigui, en cas pitjor, asimptòticament més eficient que el codi anterior? Si existeix, expliqueu-lo a alt nivell (no cal codi concret) i analitzeu el seu cost. Si no existeix, expliqueu per què no pot existir.

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