

Cognoms

Nom

DNI

Examen Parcial EDA

Duració: 1h 30min

31/03/2022

-
- *L'enunciat té 3 fulls, 6 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.*
 - *Lleuat que es digui el contrari, sempre que parlem de cost ens referim a cost asimptòtic en temps.*
 - *Lleuat que es digui el contrari, cal justificar les respostes.*
-

Problema 1

(6 pts.)

Responen les preguntes següents:

(a) (1.5 pts.) Considereu el codi següent:

```
int n; cin >> n;
vector<int> v(n);
for (int i = 0; i < n; ++i) v[i] = i+1;
random_shuffle(v.begin(), v.end()); // Theta(n)
int s = 0;
for (int i = 0; i < n; ++i)
    for (int j = 0; j < v[i]; ++j) ++s;
cout << s << endl;
```

Recordem que, donat un vector v , la instrucció `random_shuffle(v.begin(), v.end())` reordena els elements del vector v de manera aleatòria en temps lineal en la mida del vector. En funció d' n , què calcula el codi anterior i quin és el seu cost asimptòtic?

(b) (2 pts.) Considereu ara el codi següent:

```
int n; cin >> n;
for (int j = 1; j < n; ++j){
    int k = 2;
    while (k < n) k = k * k;
}
```

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

Cognoms

Nom

DNI

- (c) (2.5 pts.) Per a qualsevol nombre natural $n \geq 1$, definim el vector de naturals següent:

$$(1, 2n, 2, 2n-1, 3, 2n-2, 4, 2n-3, \dots, n, n+1)$$

En funció d' n , quin és el cost de l'algorisme d'ordenació per inserció sobre aquest vector?

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

Cognoms

Nom

DNI

Problema 2

(4 pts.)

Donat un vector v d' n naturals diferents i ordenats de forma creixent, volem saber quin és el natural més petit que no apareix a v . Recordem que el 0 és un natural.

- (a) (1.5 pts.) Contactem amb un amic que és ben conegut per proporcionar solucions estranyes i innecessàriament complicades, i ens comenta que la funció *inefficient* soluciona aquest problema:

```
bool find (int x, const vector<int>&v, int pos){  
    if (pos < 0) return false;  
    return v[pos] == x or find(x,v,pos-1);  
}
```

```
int inefficient (const vector<int>& v) {  
    int n = v.size ();  
    for (int i = 0; i < n; ++i)  
        if (not find(i,v,n-1)) return i;  
    return n;  
}
```

En funció d' n , quin és el cost en cas pitjor d'una crida a la funció *inefficient*?

- (b) (2.5 pts.) Doneu vosaltres una funció que solucioni aquest problema i que triqui, en cas pitjor, $\Theta(\log n)$.

```
int efficient (const vector<int>& v, int l, int r) {
```



```
}
```

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