

Cognoms**Nom****DNI****Examen Parcial EDA****Duració: 2h****21/04/2023**

-
- *L'enunciat té 4 fulls, 8 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**(4.5 pts.)**

Donats dos vectors de nombres naturals v_1 i v_2 de mida $n > 0$, volem determinar si podem trobar un element a cada vector tal que, si els intercanviem, els dos vectors resultants sumen el mateix. Per exemple, si $v_1 = (6, 4, 3, 9)$ i $v_2 = (7, 0, 2, 5)$, aleshores intercanviant el 6 i el 2 obtenim dos vectors que sumen el mateix. En canvi, si $v_1 = (2, 0, 9, 5)$ i $v_2 = (2, 4, 5, 7)$ no existeix cap parell d'elements amb la propietat desitjada.

(a) (1 pt.) Considereu la solució següent al problema plantejat:

```
int suma(const vector<int>& v){
    int s = 0;
    for (int x : v) s += x;
    return s;
}

pair<int,int> sol_facil (vector<int>& v1, vector<int>& v2) {
    for (int i = 0; i < v1.size (); ++i)
        for (int j = 0; j < v2.size (); ++j) {
            swap(v1[i], v2[j]);
            int s1 = suma(v1);
            int s2 = suma(v2);
            swap(v1[i], v2[j]);
            if (s1 == s2) return {v1[i], v2[j]};
        }
    return {-1, -1}; // No hi ha solució
}
```

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

- (b) (2 pts.) Completeu el codi següent per tal que sigui una solució vàlida al problema plantejat:

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

pair<int,int> sol (const vector<int>& v1, const vector<int>& v2) {
    int dif = suma(v2) - suma(v1); // suma és la funció de l'apartat anterior
    if (dif%2 != 0) return {-1,-1};
    for (int i = 0; i < v1.size (); ++i) {
        int x = 
        if (cerca(v2, x, 0, v2.size()-1)) return {v1[i], x};
    }
    return {-1,-1};
}
```

Analitzeu, en funció d' n , el cost en el cas pitjor d'una crida a *sol*.

Cognoms

Nom

DNI

- (c) (1.5 pts.) Expliqueu com modificaríeu la solució de l'apartat anterior per tal que el seu cost, en el cas pitjor, sigui millor asimptòticament. No cal que doneu codi concret, una descripció a alt nivell serà suficient. Quin és el cost, en el cas pitjor, de la nova solució?

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

Cognoms

Nom

DNI

Problema 2

(5.5 pts.)

Donat dos nombres naturals n, k diferents de zero, volem expressar n com la suma ordenada d'exactament k potències de 2 amb exponent no negatiu (és a dir, 2^{-3} no la considerem una potència de 2 vàlida). Per exemple, si $n = 21$ i $k = 4$, una solució és $21 = 2^3 + 2^3 + 2^2 + 2^0$. Fixem-nos que l'ordre desitjat és decreixent.

Observació: la representació en binari d' n ens dona una manera d'expressar n com a suma de potències de 2, però pot no tenir exactament k sumands. De fet, és la representació amb el menor nombre de sumands possible.

- (a) (0.75 pts). Escriviu solucions per a $n = 10$ i $k = 2, 3, 4$ i 5 . No cal justificar com les obteniu.

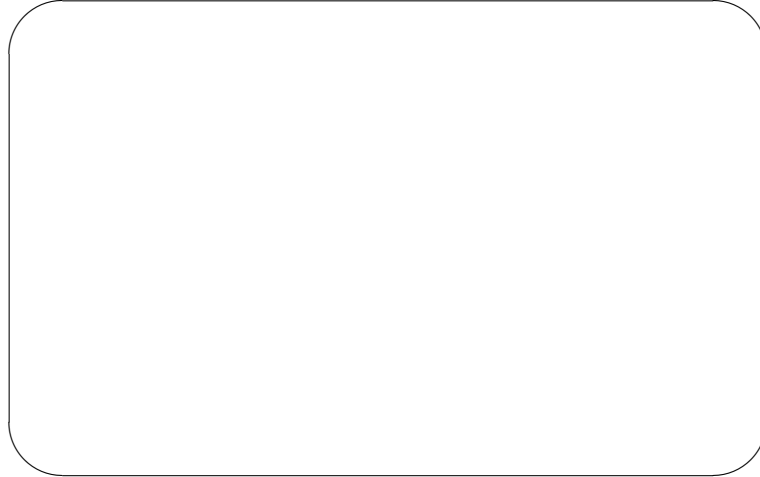
- (b) (1.25 pts.) Quines són les dues úniques situacions on no hi ha solució?

- (c) (1.5 pts.) Completeu la funció següent per tal que donat un natural n retorni, ordenades de menor a major, les posicions de la representació en binari d' n on apareix un 1. Per exemple, si $n = 19$, cal retornar el vector $(0, 1, 4)$.

```
vector<int> pos_uns (int n) {
```

```
    vector<int> v;
```

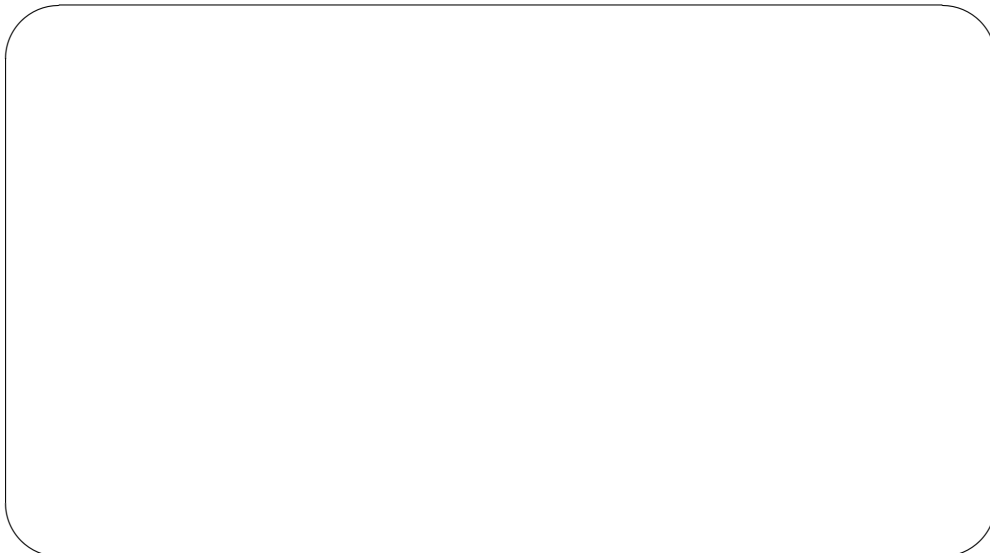
```
    int pos = 0;
```



```
    return v;
```

```
}
```

Quin és el seu cost en funció d' n ?



Cognoms

Nom

DNI

(d) (2 pts.) Completeu el codi següent per tal que resolgui el problema plantejat:

```
void escriu_suma_potencies (int n, int k) {  
    vector<int> uns = pos_uns(n);  
    if (  )  
        cout << "No hi ha solucio" << endl;  
    else {  
        priority_queue<int> Q;
```

```
        bool primer = true;  
        while (not Q.empty()){  
            if (not primer) cout << " + ";  
            else primer = false;  
            cout << "2^" << Q.top();  
            Q.pop();  
        }  
        cout << endl;  
    }  
}
```

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