

Grup G4

- Ricardo López
- María Montalvo
- Víctor Pla
- Sven Wallin

Exercici 1.6

Explicació

La primera observació que hem de fer és la quantitat de celebritats que pot haver-hi en un graf, el qual pot ser 0 o 1 celebritats. Això és degut al fet que no pot a ver-hi dues o més persones que no coneguin a ningú, però que tothom les conegui.

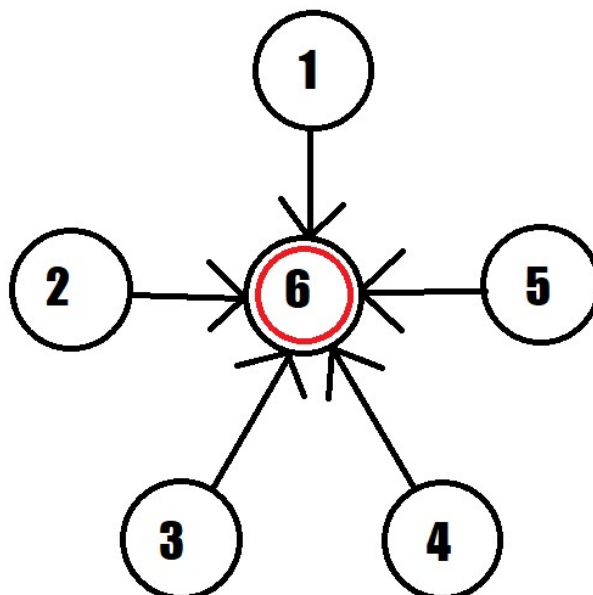
Després, donat que ha de funcionar en temps lineal $O(n)$, ens donem compte que es pot consultar la relació entre 2 nodes en ambdós sentits i sempre descartarem, com a mínim, un dels 2 nodes (*veure correctesa*).

Finalment, un cop ens quedi un possible node a comprovar, farem una verificació de si aquest és celebritat, seguint la seva definició.

Correctesa

El motiu pel qual sabem que l'algorisme funciona és perquè, cada cop que descartem un node, sabem amb totalitat que aquest no serà celebritat; és a dir, part de la solució.

Exemple de graf amb una celebritat



Codi

```
vector<int>nodes(n);
nodes=emplenar_nodes(n);
bool adj_matrix[n][n];           # Matriu d'adjacències donada a l'enunciat
int c_id = 0;
for(int i = 1; i < nodes.size(); i++){
    if(adj_matrix[c_id][i])      # Es consulta si la possible celebritat coneix
                                # a l'altre node a la matriu d'adjacències
        c_id = i;

    else if(!adj_matrix[i][c_id]) # Descartem els dos nodes si l'altre node
                                # NO coneix a la possible celebritat
                                # (sabem que c_id no coneix a i,
                                # llavors no es coneixen entre ells)
        c_id = i + 1;

}
for (i=0; i<=n; i++){
    if(c_id != i && (adj_matrix[c_id][i] || !adj_matrix[i][c_id])){
        c_id = -1;
        break;
    }
}
if(c_id != -1)
    cout << "Celebritat trobada: " << c_id << endl;
else
    cout << "No hi ha cap celebritat al graf" << endl;
return 0;
```

Cost de l'algorisme

$O(n) + O(2n) + O(2n) = O(5n) = O(n) \rightarrow$ LINEAL