

N nodos
 $N = 5$
 M aristas
 M



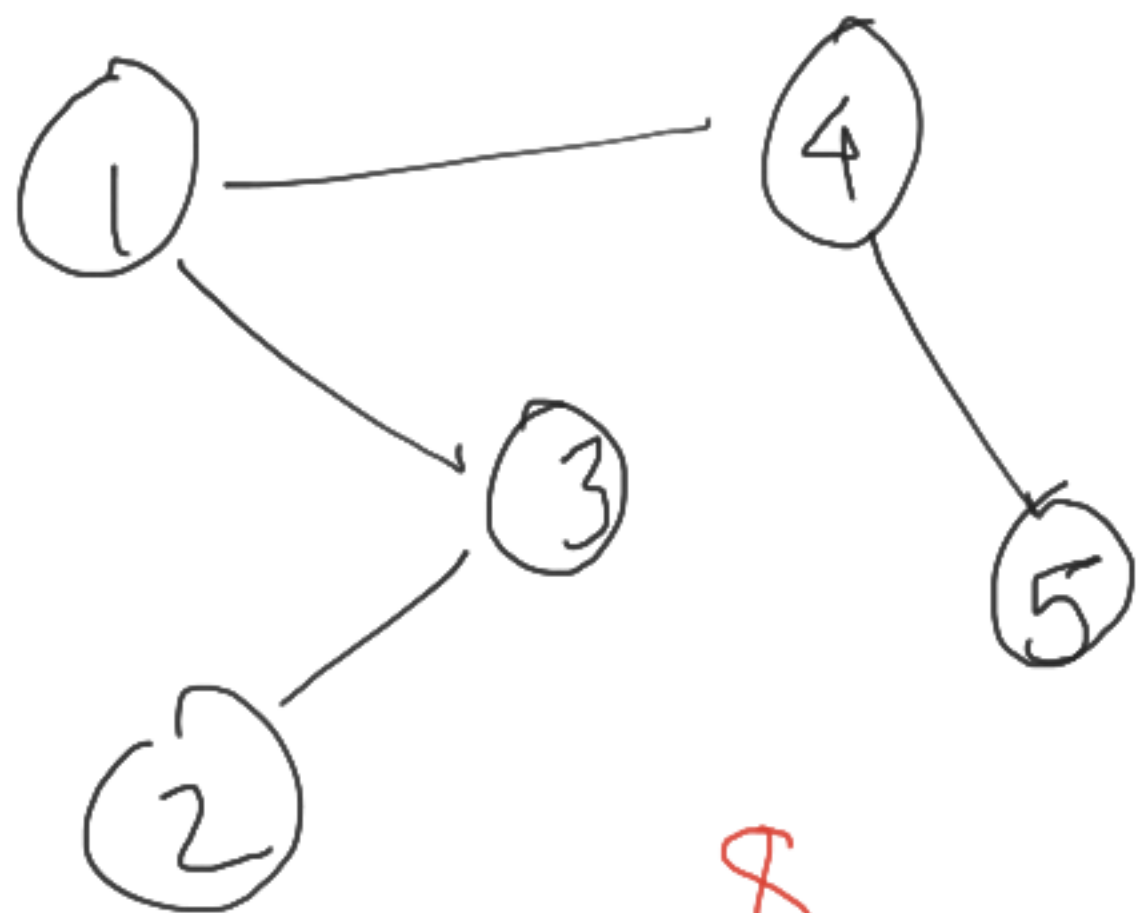
$5 + 8 = 13$
 $N + M = \text{Total}$



$O(N) + O(M)$

Complejidad Final
 $O(N + M)$

$O(V + E)$



8

5 nodos

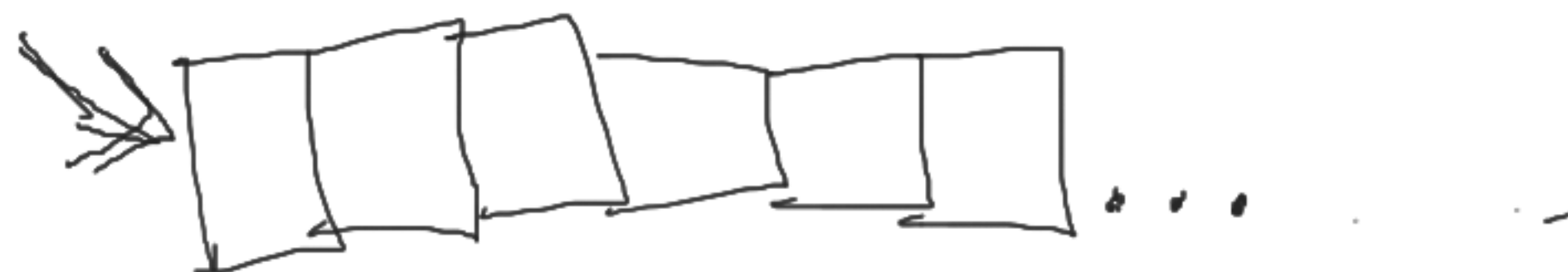
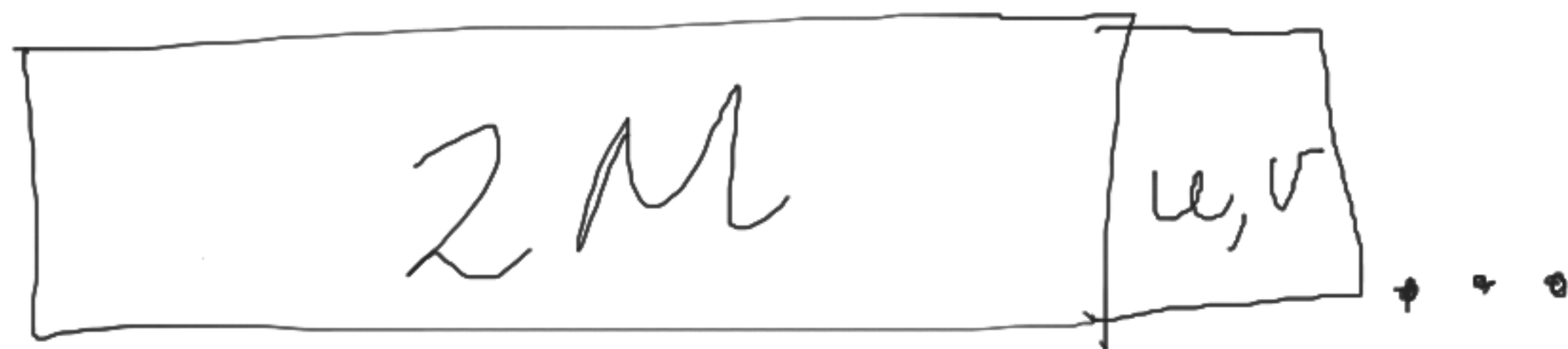
(1, 4) (4, 5)
(4, 1) (5, 4)
(1, 3)
(3, 1)
(3, 2)
(2, 3)

size?

M

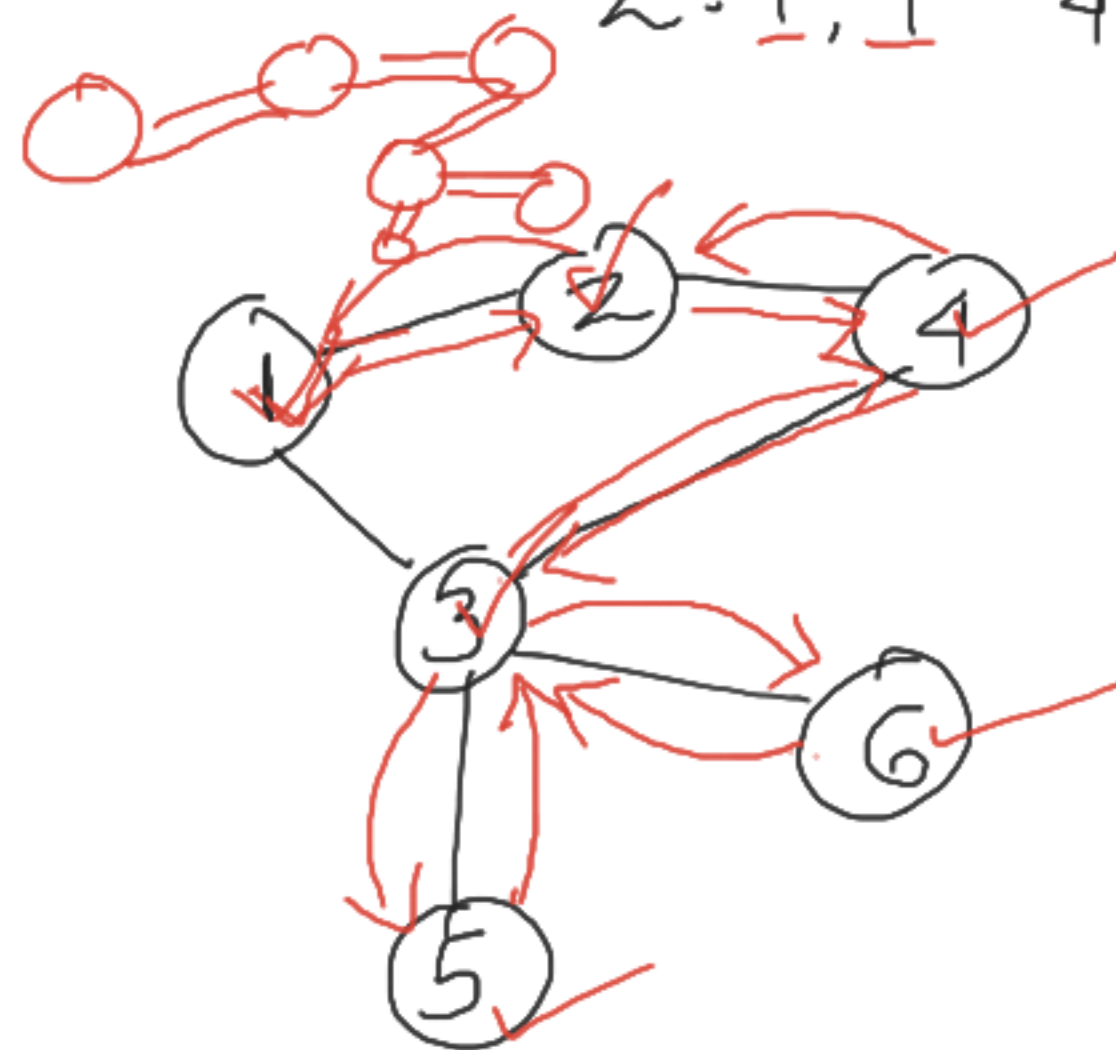
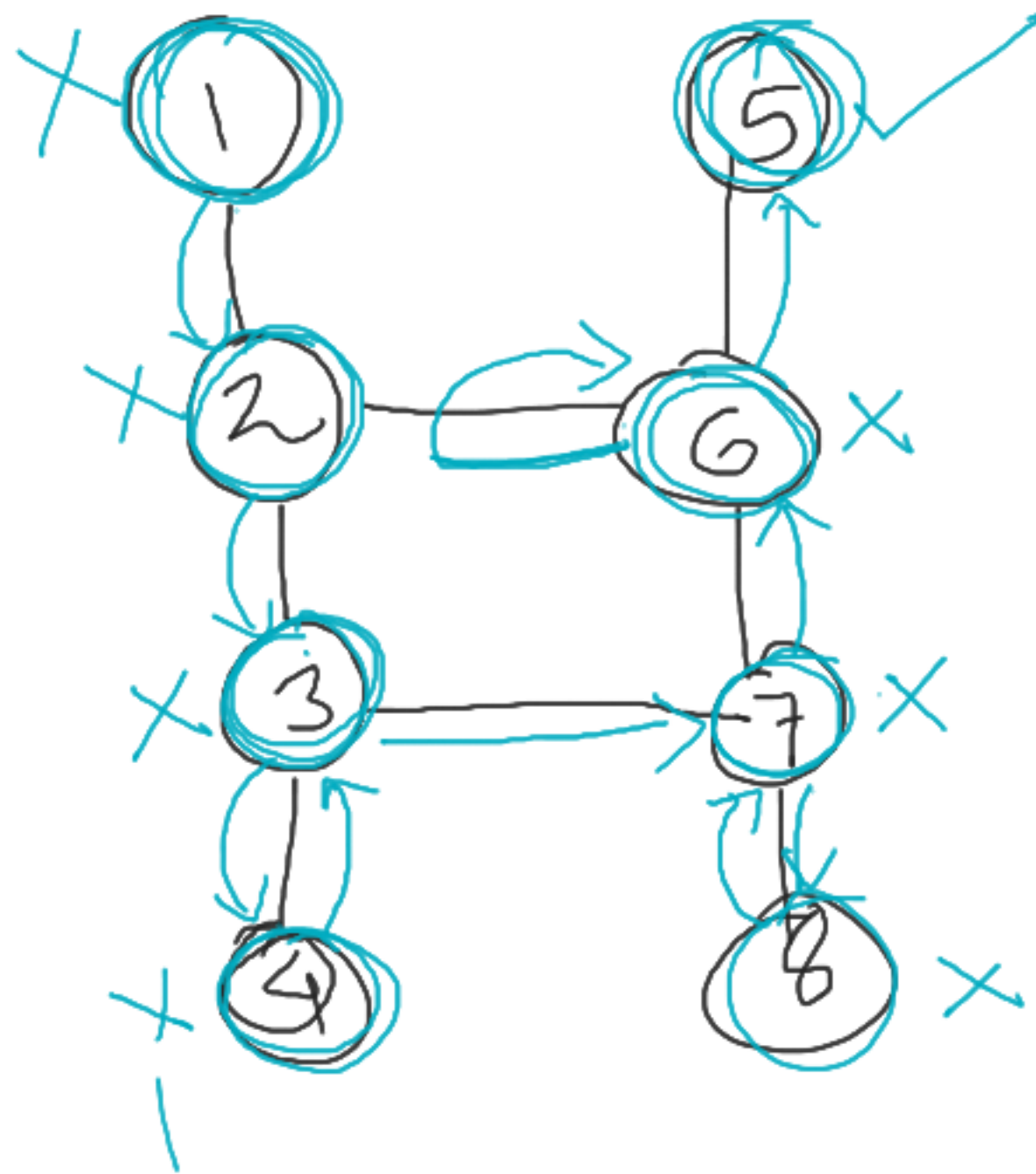
2M

$\Theta(M)$



DFS \rightarrow Profundidad

weiburg
↓ ↓ ↓
1: 2, 3 3: 1, 4, 5, 6
2: 1, 4 4: 2, 3
5: 3
6: 3

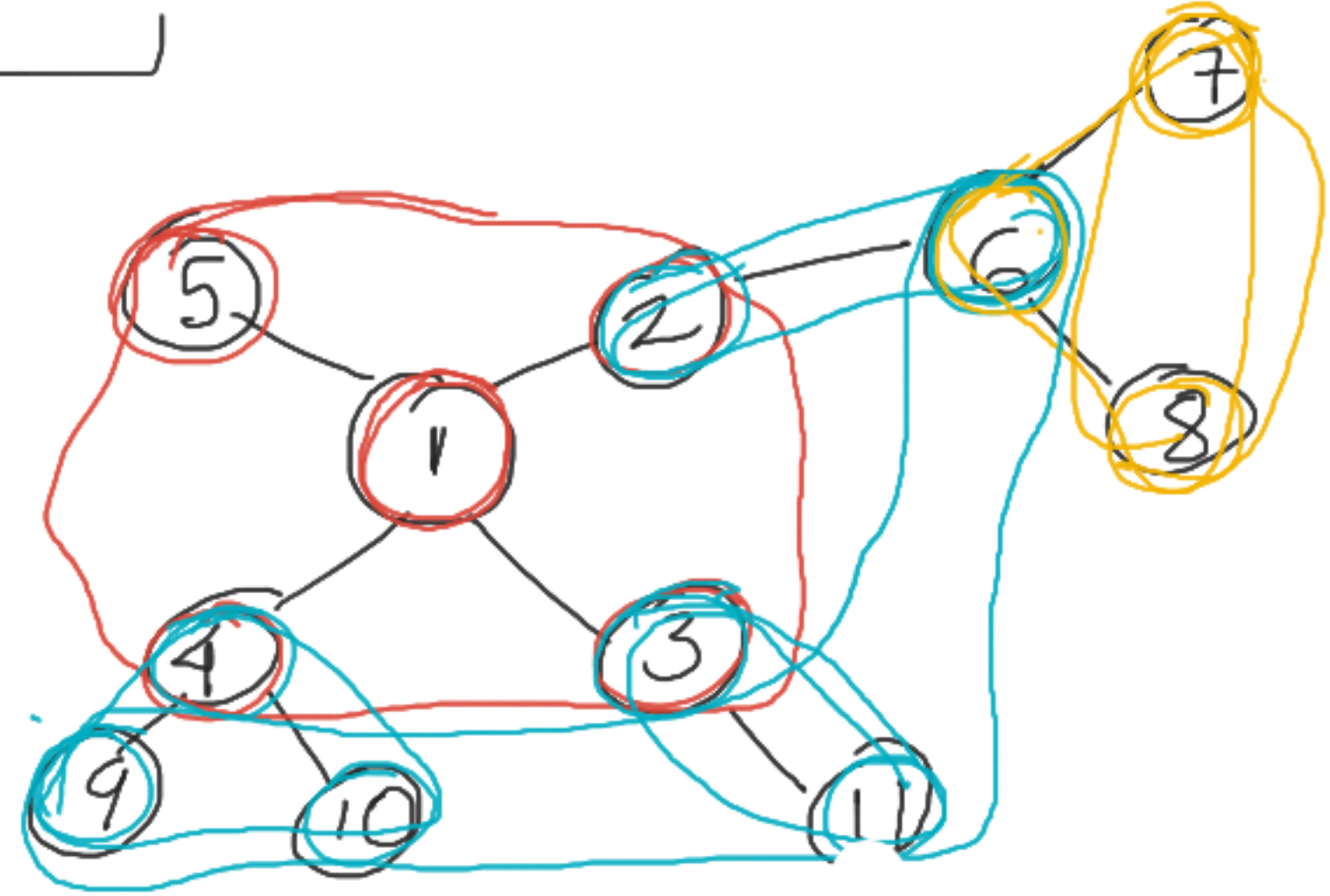
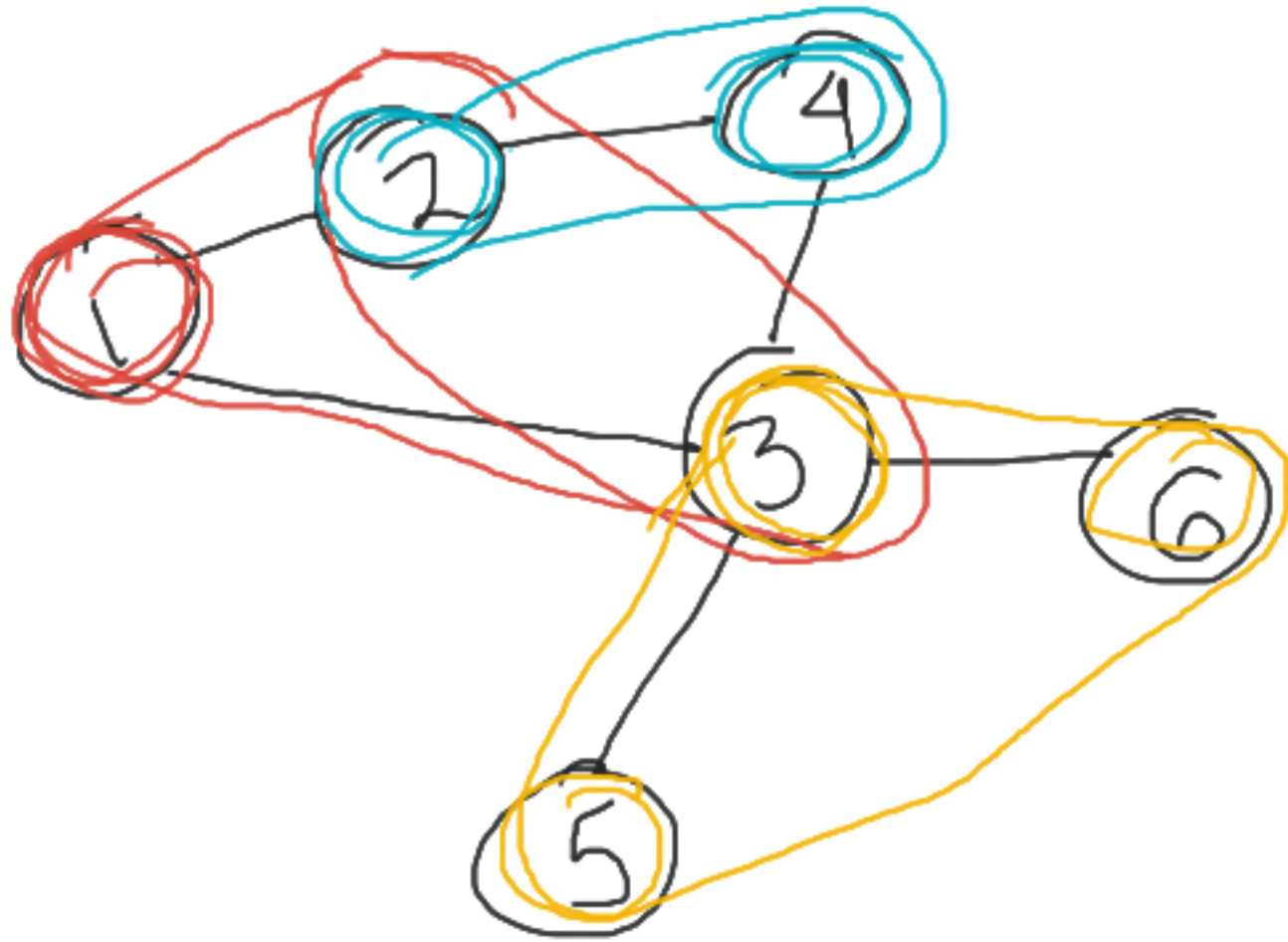


BFS → Capas, anchura

1: 2, 3
2: 1, 4
5: 3

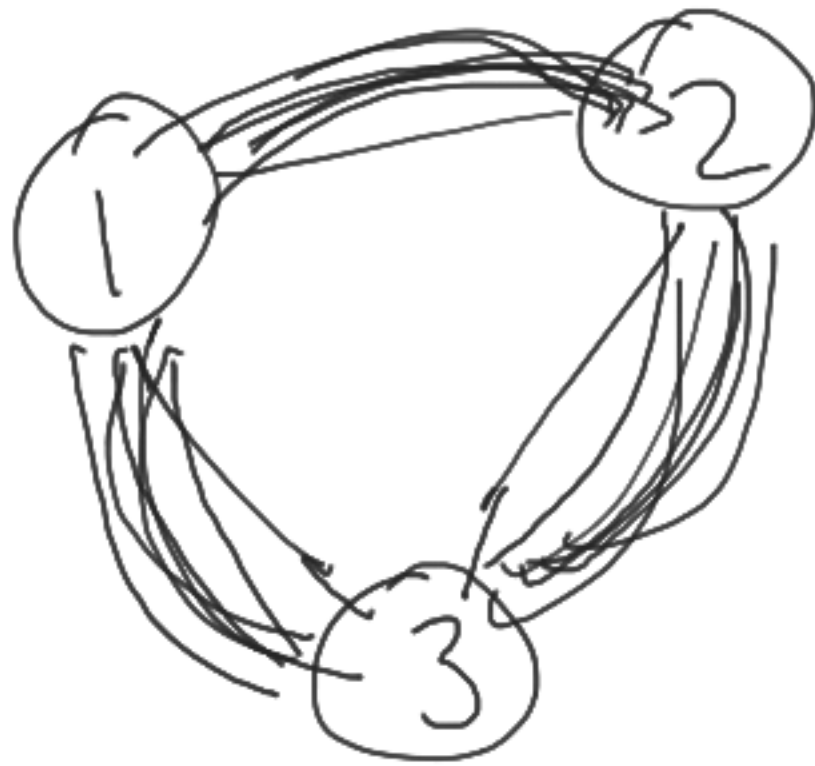
3: 1, 4, 5, 6
4: 2, 3
6: 3

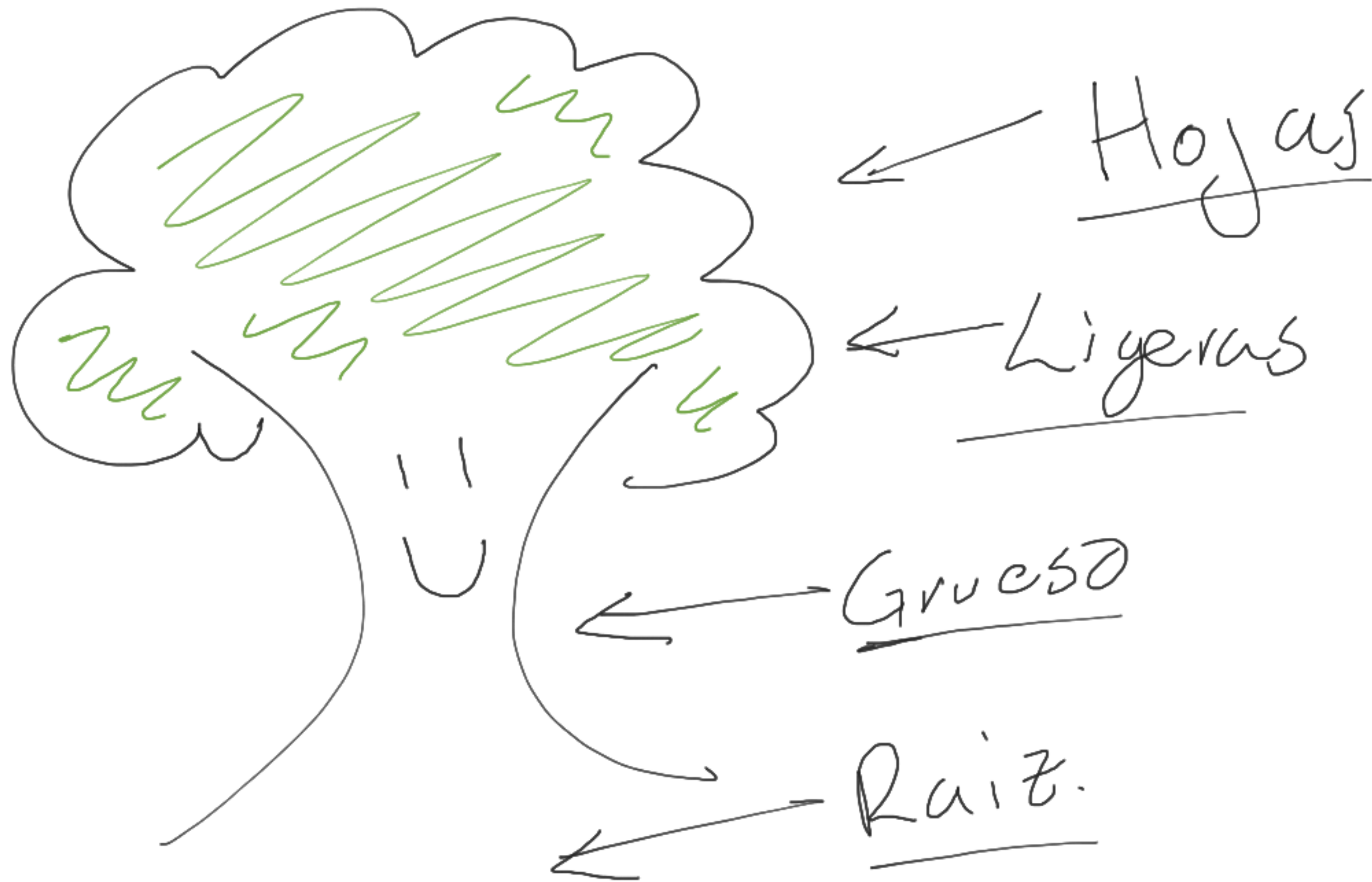
→ $\frac{4}{2}$

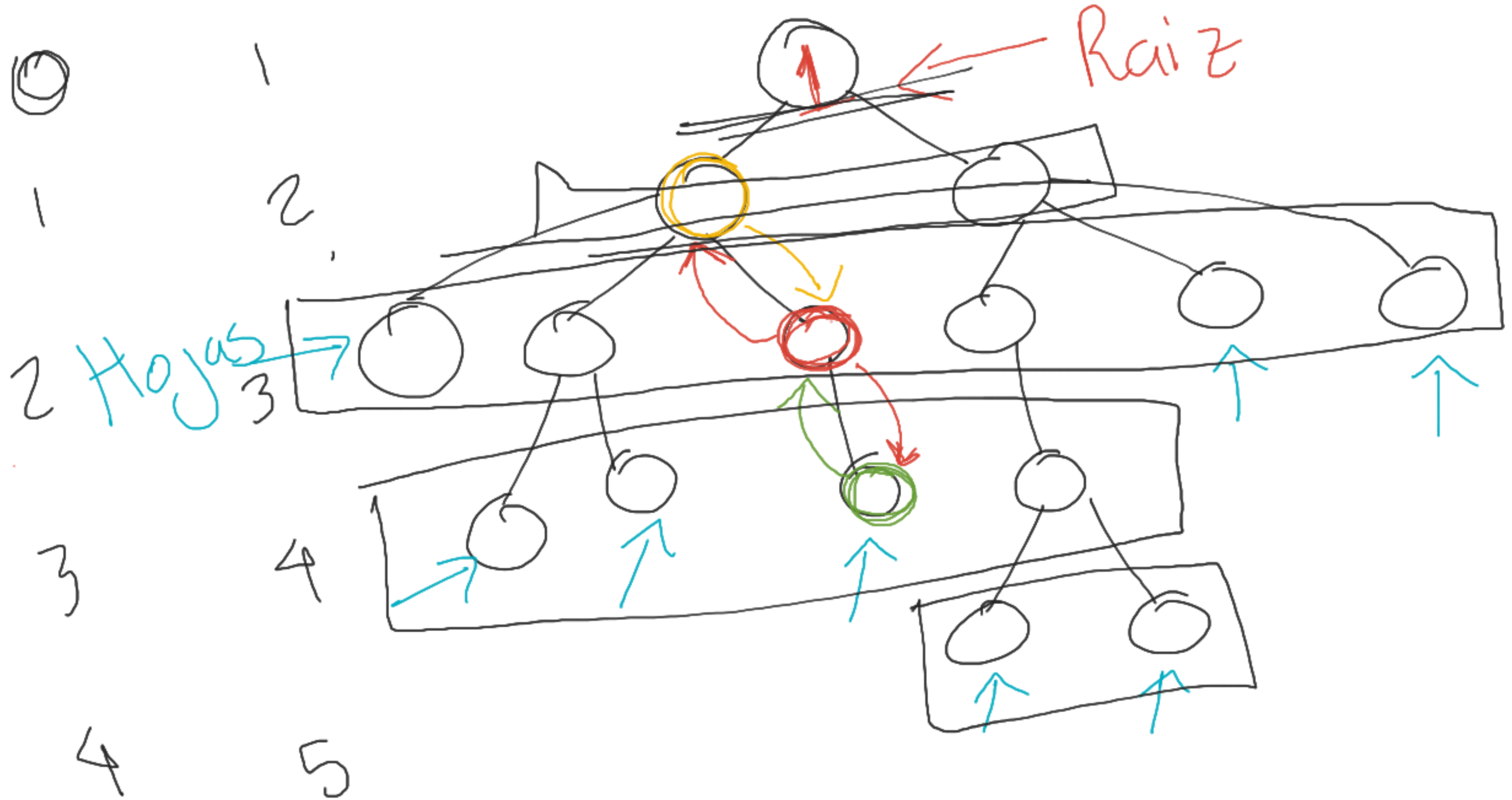


ARBOLES

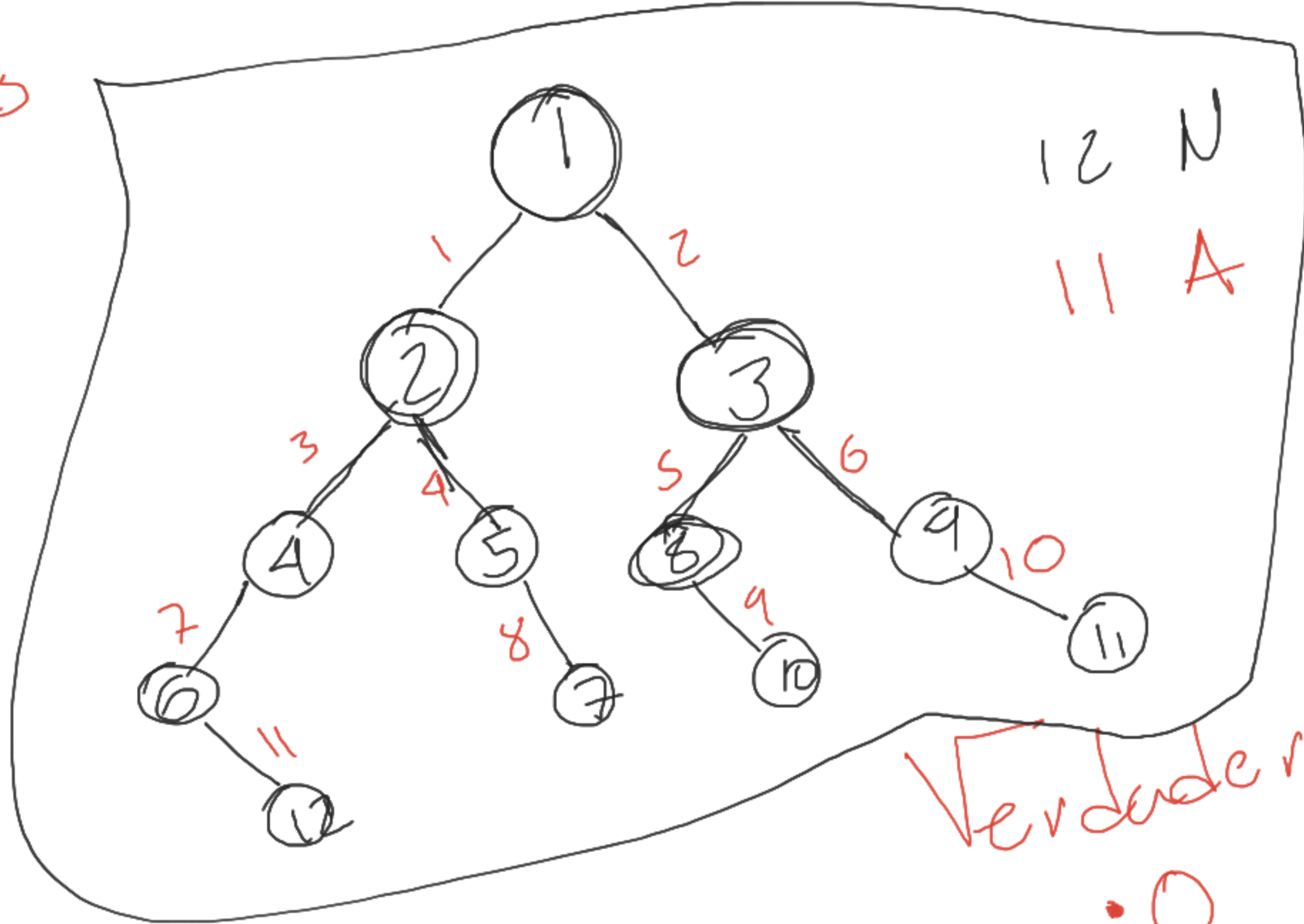
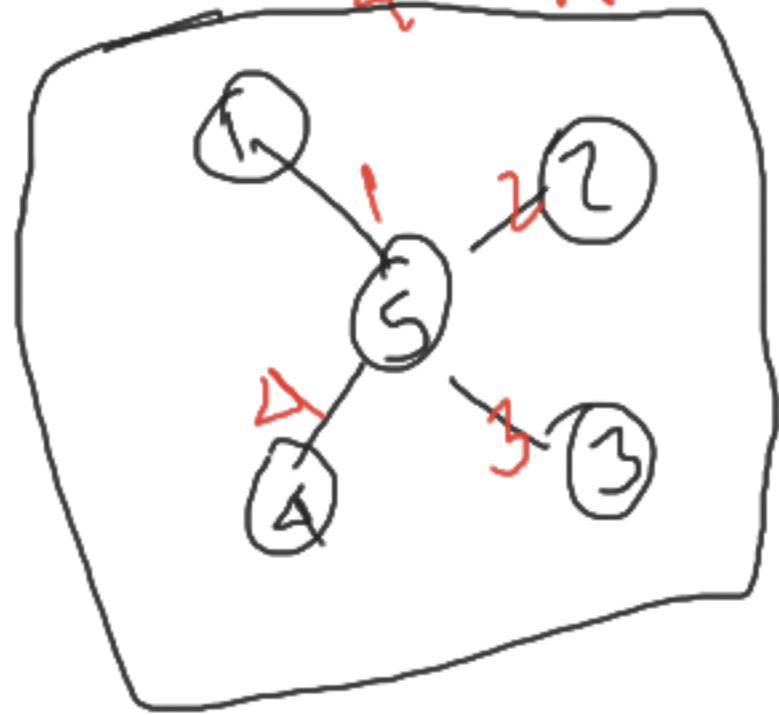
Tipo de grafo \rightarrow 3 Propiedades







5 No 205
Avidas



12 N
11 A

Verdadero
:0

$$\begin{array}{c} 3 \\ \swarrow \downarrow \searrow \\ 2 \quad 3 \quad 2 \\ \hline \hline \end{array}$$

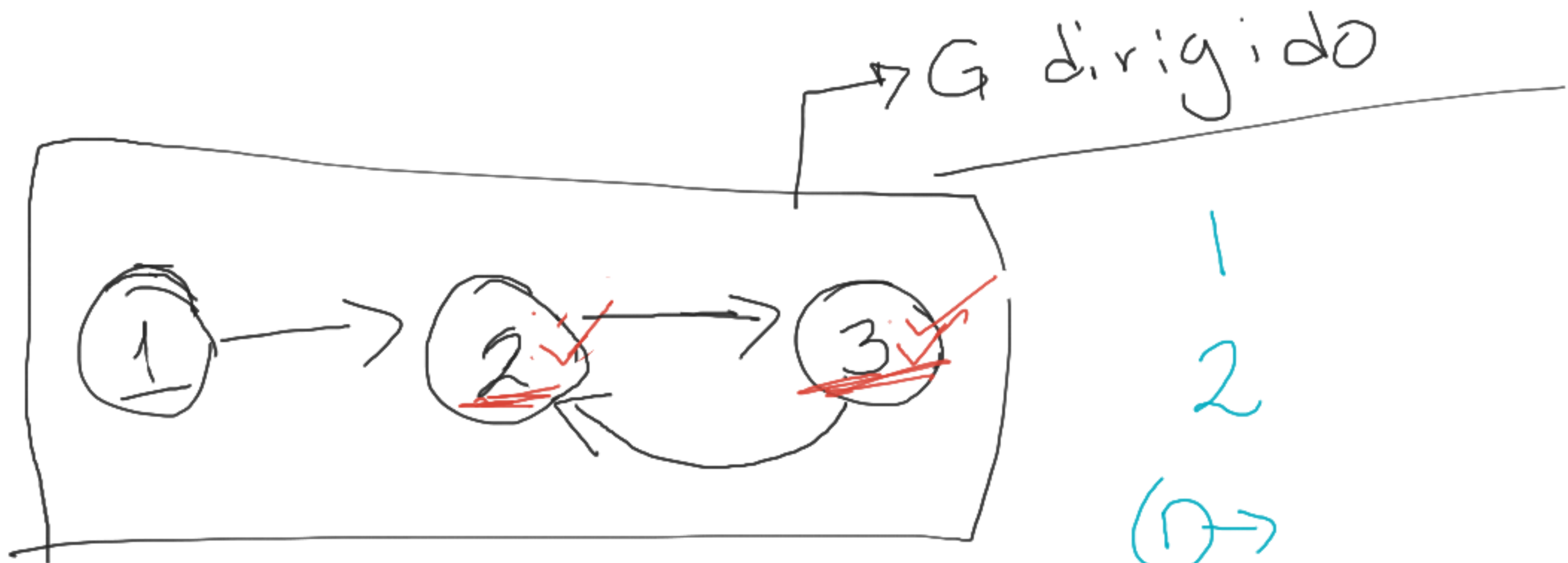
$a = 1$
 $res = 2$

$a = 2$
 $res = 2$

$a = 3$
 $res = 3$

$$\begin{array}{r} 2 \quad 2 \quad 3 \\ \hline 4 \end{array}$$

$$\begin{array}{c} 3 \\ 1 \quad 2 \quad 3 \end{array}$$



$$\begin{array}{c} 1 \\ 2 \\ \hline 1 \quad 2 \quad 3 \end{array}$$

4
 4 2 3 4 1

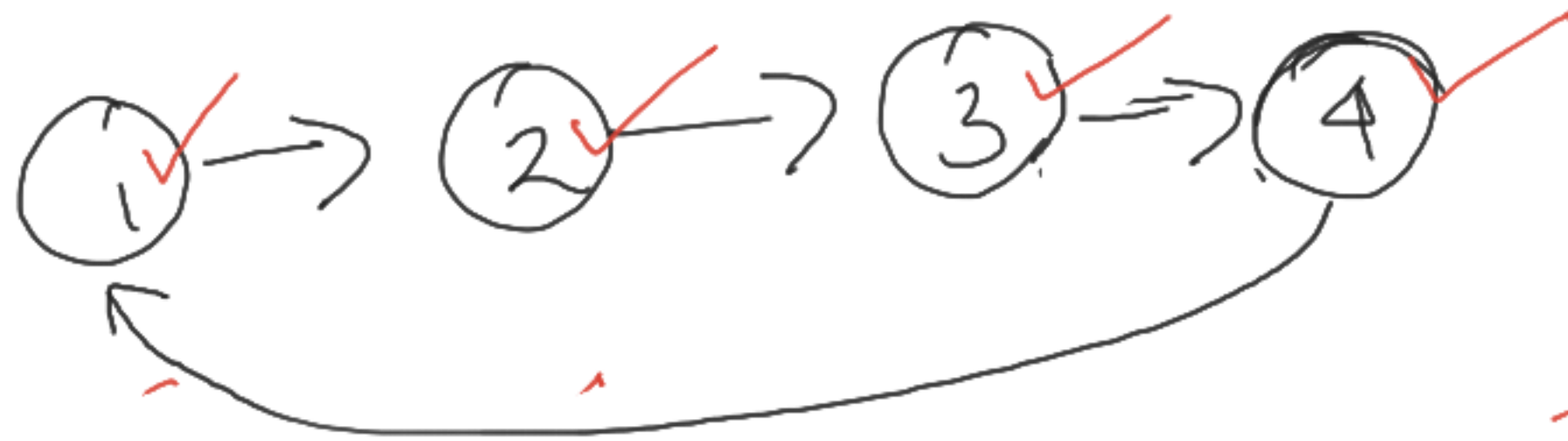
arr = [t, t, t, t]
vis

a = 1

1 = La que se pas

L > 1

2 = DFS



DFS
 ↳ Rapido de codcar

1 2 3 4

3 3 3 4

BFS

```
int DFS (int nodo_a_procesar){  
  
    vis[nodo_a_procesar] = true;  
    int res;  
    for(int i = 0; i < lista[nodo_a_procesar].size(); i++){  
  
        int nodo_siguiete = lista[nodo_a_procesar][i];  
        if(!vis[nodo_siguiete]){ //Si no esta visitado, lo visito  
            return DFS(nodo_siguiete);  
  
        } else { // Si esta visitado, quiere decir que ya encuentre a quien me voy a torcer xD  
            //return DFS(nodo_siguiete);  
        }  
    }  
  
    return res;  
}
```

~~10x~~
[t, t, t]
1 2 3

DFS(3)

3 4, 5



DFS(1) -> 2

1 . . . 10
10 1
Misma def.



1 1 1	res = 10
?	res
	res
	nodo

