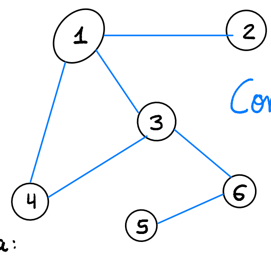
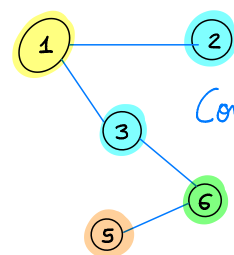
1.14. Es diu que un vèrtex d'un graf connex és un *punt d'articulació* del mateix si, en suprimir aquest vèrtex i totes les arestes que hi incideixen, el graf resultant deixa de ser connex. Per exemple, un graf en forma d'anell no té cap punt d'articulació mentre que tot node que no sigui fulla d'un arbre és punt d'articulació. Dissenyeu un algorisme que, donat un graf connex no dirigit $G = (V, E)$, indiqui quins vèrtexs del graf són punts d'articulació. Calculeu el seu cost. Indiqueu quina implementació del graf és la què proporciona un algorisme més eficient.



Connex



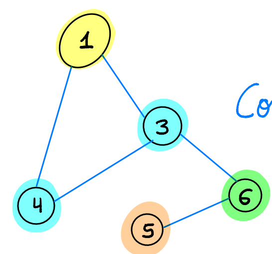
Connex

Llista Adjacència:

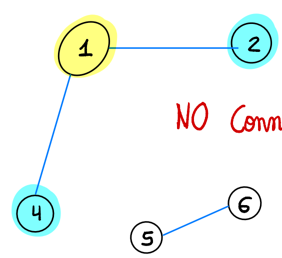
1	2	3	4
2	1		
3	1	4	6
4	1	3	
5	6		
6	3	5	

Matriu d'adjacència:

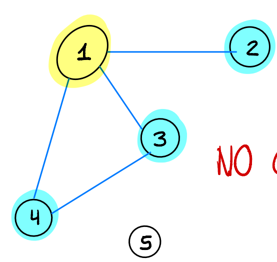
$$\begin{array}{c|c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \left(\begin{array}{cccccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right) \end{array}$$



Connex



NO Connex $\Longrightarrow$ 3 és Punt d'Articulació



NO Connex $\Longrightarrow$ 6 és Punt d'Articulació

Algorisme:

Per a tot v_i en V:	Matriu	Llista
• eliminar v_i de V i totes les seves arestes.	$O(n)$	$O(1)$
• DFS sobre $G = (V - \{v_i\}, E - \{w_{i_1}, w_{i_2}, \ldots\})$	$O(n+m)$	$O(n+m)$
• si no hem visitat tots els nodes $v \in V$, v_i és un punt d'articulació	$O(1)$	$O(1)$
• restaurem v_i a V i totes les seves arestes.	$O(n)$	$O(1)$

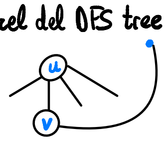
$$O(n(n+m)) = O(n^2 + nm) \underset{n \leq m}{=} O(nm)$$

IDEA CLASSE:

DFS modificat —— $O(n+m)$
(DFS Tree)

$u \in V$ és punt d'articulació $\Longleftrightarrow$

- si u és fulla $\longrightarrow$ no és punt d'articulació
- si u és arrel del DFS tree i té més d'un fill —— és punt d'articulació
- si u no és arrel del DFS tree i no passa:



que d'un fill es pugui anar a un node superior —— és punt d'articulació