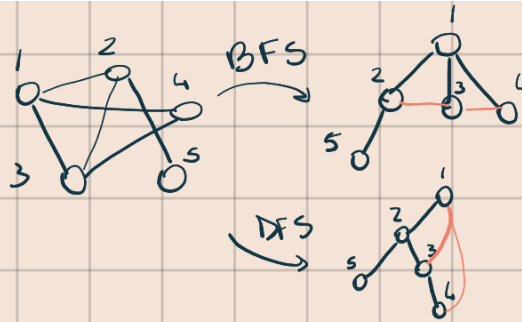
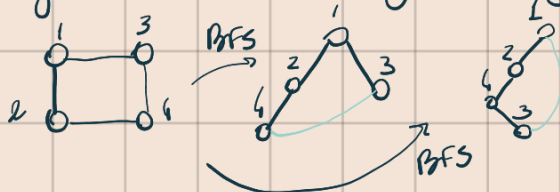


18. Dado un grafo G queremos responder una serie de consultas de la siguiente forma: dados dos vértices v y w , determinar si existe un único camino entre v y w . Diseñar un algoritmo que dado un grafo G lo procese para generar una estructura de datos que permita responder cada consulta en $O(1)$ tiempo. El mejor algoritmo que conocemos toma tiempo $O(n + m)$.

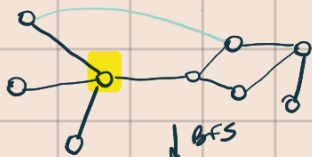
Supongamos



Dados v y w si v es padre de w en el árbol BFS y no tengo un backedge ni cross edge que salga de w el camino es único. Si v es k -ancestro de w con $k \geq 2$ y en algún nivel intermedio hay cross-edge o backedge hay mas de uno



o si forman parte de \perp

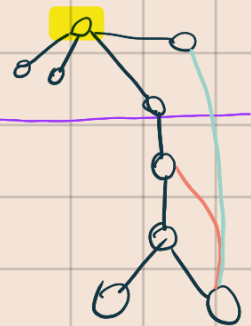


Si hay \perp ciclo entre v y w hay \oplus de un camino
Si no hay un único



a partir de
aquí
deja de haber un
único camino
para llegar a los
vértices después
de v

DFS



Casos:

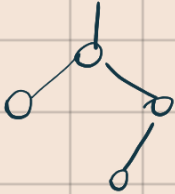
- v y w pertenecen a un ciclo $\rightarrow \exists!$
- v es el padre de w y w NO pertenece al mismo ciclo que el padre o NO pertenece a ningún ciclo $\rightarrow \exists!$
- v y w NO tienen un ciclo de por medio $\rightarrow \exists!$
- v y w tienen un ciclo de por medio $\rightarrow \exists!$
- v es padre de w y están en distintos ciclos $\rightarrow \exists!$

En un árbol $\forall v, w \in V(T) \exists! P v \rightarrow w \Rightarrow$ podría aprovechar y ver los ciclos



↓
puedo colorear
el ciclo
de un color
distinto
Si distintos
ciclos
comparten
una o varias
los puntos del mismo
color

Debería hacer el árbol DFS \rightarrow Si NO hay backedges $\rightarrow \exists!$ camino $\forall v, w$
Si hay \rightarrow puede \exists más de un camino
entre un par de vértices



↓
dada v y w me debería fijar si en sus
niveles intermedios hay ciclos

Algoritmo

Hago árbol DFS de G

Si no hay backedges \rightarrow hay un camino único $\forall v, w \in V(G)$

Si hay backedges

\hookrightarrow si entre v y w NO hay ciclo $\rightarrow \exists!$ camino $P_{v \rightarrow w}$

\hookrightarrow si v y w no están en ciclos distintos y v es padre de w
viceversa NO hay único camino

\hookrightarrow si están en distintos ciclos y v es padre de $w \rightarrow \exists!$ camino

\rightarrow Para identificar ciclos dependo de los backedges \rightarrow para evitar
cross-edges y que sea + fácil DFS

\rightarrow Si entre v y w NO hay ciclo \rightarrow se saca una arista del camino
que los une \rightarrow genera 2 componentes

→ Si hay ciclos → solo una arista y puedo generar 2 componentes como NO

→ Si la arista padre de w genera 2 componentes → el camino es único ya que solo puedo acceder a w mediante ella. Si no, hay \oplus de una forma de acceder a w

Algoritmo

me devuelve un arreglo donde nos dice para q nodes su cubren
→ Si me tiene cubren 0 → su arista padre es puente

1. Uso los algoritmos para encontrar aristas puente → $O(n+m)$

2. Dados v y w me fijo en el arreglo cuál es el cubren de w . Si es 0 es único. Si no hay más de 1 $O(1)$
↓
Suponemos que el camino es de v a w