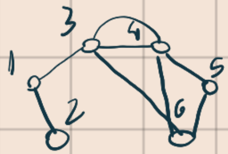
16. Un grafo G es un *cactus* cuando cada una de sus aristas pertenece a un único ciclo.

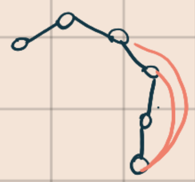
a) Sea T un árbol DFS de un grafo G y sea $T(v,w)$ el unico camino entre v y w en T para todo $v, w \in V(G)$. Demostrar que G es un cactus si y solo si para toda arista $vw \in E(G) \setminus E(T)$ ocurre que $T(v,w) + vw$ es el único ciclo que contiene a las aristas en $T(v,w)$.

$\leftarrow$) Si $\forall vw \in E(G) \setminus E(T)$ ocurre que $T(v,w) + vw$ es el único ciclo que contiene a las aristas en $T(v,w)$ QVQ G es cactus

Si $\forall vw \in E(G) \setminus E(T) \rightarrow T(v,w) + vw$ es el único ciclo que contiene a las aristas de $T(v,w)$ en $G \rightarrow$ todos los ciclos de G tienen aristas distintas $\rightarrow G$ es cactus

$\rightarrow$) Si G es cactus QVQ $\forall vw \in E(G) \setminus E(T)$ $T(v,w) + vw$ es el único ciclo que contiene a las aristas de $T(v,w)$

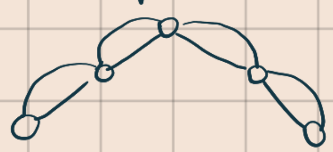




Sup que NO $\rightarrow \exists vw \in E(G) \setminus E(T) / T(v,w) + wv$ NO es el único ciclo que contiene a las aristas de $T(v,w) \rightarrow \exists ab / T(a,b) \subseteq T(v,w)$ ($ab \neq vw$ pero $a = v$ o $b = w$) y tq $T(a,b) + ba$ es un ciclo $\rightarrow \exists$ 2 ciclos que comparten aristas en $G \rightarrow G$ NO es cactus

b) Demostrar que los grafos cactus tienen $O(n)$ aristas.

En esencia, por cada nodo podemos tener una sola backedge



El árbol DFS de un grafo cactus G va a tener $n-1$ aristas QVQ cada nodo v puede tener a lo sumo 1 backedge que lo lleve a w y w tiene una backedge lo lleva a v.

Si este NO fuera el caso $\rightarrow G$ tendría 2 ciclos que comparten al menos 1 arista $\rightarrow$ NO sería cactus

c) Diseñar un algoritmo de tiempo $O(n)$ para determinar si un grafo es un cactus. En caso afirmativo, el algoritmo debe retornar todos los ciclos del grafo. En caso negativo, el algortimo debe retornar dos ciclos que compartan una arista.

Podriamos aprovecharnos de que la complejidad del DFS/recorrerlo es $O(n)$

1. Hago DFS y Cubren(G)
2. Recorro cubren y me fijo si algún nodo tiene cubren ≥ 2
3. Si tiene recorro el árbol y devuelvo 2
4. Si no devuelvo todos los ciclos

d) Diseñar un algoritmo de tiempo $O(n)$ para encontrar un árbol generador mínimo de un grafo cactus. **Justificar** que el algoritmo es correcto utilizando resultados conocidos.

1. Busco todos los ciclos $O(n)$
2. Para c/ciclo elijo la arista de mayor peso y la marco (si hay + de una cualquiera)
3. Hago DFS y me aseguro que NO se agregue lo marcado

e) Proponer una formula para contar la cantidad de árboles generadores mínimos de un grafo cactus que pueda ser computada en $O(n)$ operaciones de suma y multiplicación. **Demostrar** que la fórmula es correcta.

Por c/ciclo del grafo podemos armar tantos AGMs como aristas de máximo valor haya. Sup que $\exists$ a y b/ $c(a) = c(b) = C_{max}(C)$ con C ciclo $\rightarrow$ tenemos 2 posibilidades para el AGM

↳ esto va a ser así por c/ciclo que tengamos

$$\#\text{AGMs Diferentes} = \prod_{i=1}^{\#\text{ciclos en } G} \sum_{\substack{vw \in C_i / \\ c(vw) = C_{max}(C_i)}} vw$$