Clase 03 MA3705. 07 de septiembre 2020.

Tabla

- Generación de aristas
- Algoritmos de búsqueda
- Problema del árbol/bosque generador de peso mínimo (MST).

Generación de aristas

Generación (span) de aristas

Sea $G = (V, E)$ un grafo, $F \subseteq E$.

- Definimos el generado

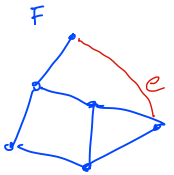
$$\text{span}(F) = \{e = uv \in E : \exists u\text{-}v \text{ camino en } F\}$$

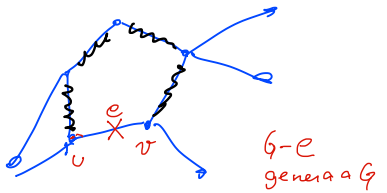
- Decimos que H (o $E(H)$) genera G si $E(G) \subseteq \text{span}(E(H))$.

Lemas

Si H genera a G entonces cada componentes de G está contenida en una componente de H.

Si $e = uv \in E(G)$ pertenece a algún ciclo de G entonces $G - e$ genera G.





Abuso de notación

Sea $F \subseteq \binom{V}{2}$.

Decimos que F es árbol, bosque, camino, ciclo, etc.

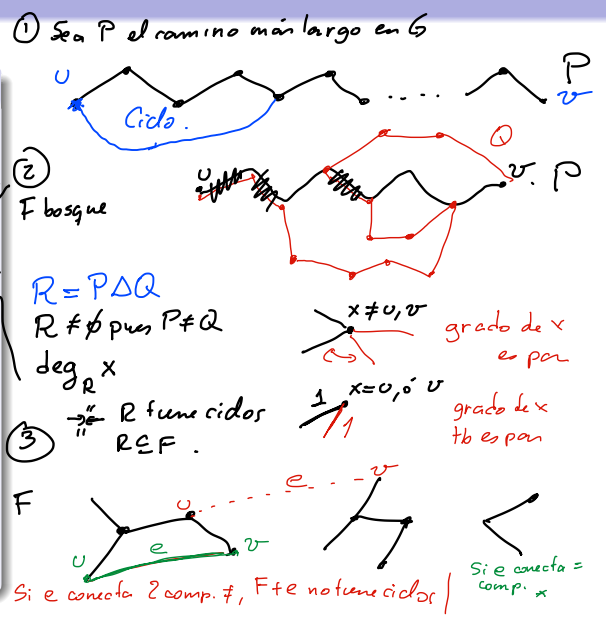
cuando (V, F) es árbol, bosque, camino, ciclo, etc.

Resultados útiles

Sea $G = (V, E)$ grafo $F \subseteq E$, $e \in E$.

Lemas

1. Si (V, F) tiene alguna arista, y todos los grados son pares, entonces tiene un ciclo.
2. Si F bosque entonces para cada $u, v \in V$ existe a lo más 1 u-v camino.
3. Si F bosque entonces $F + e$ tiene a lo más un ciclo $C(F, e)$.
4. Si F bosque generador entonces (V, F) tiene las mismas componentes que G y cada una es un árbol.

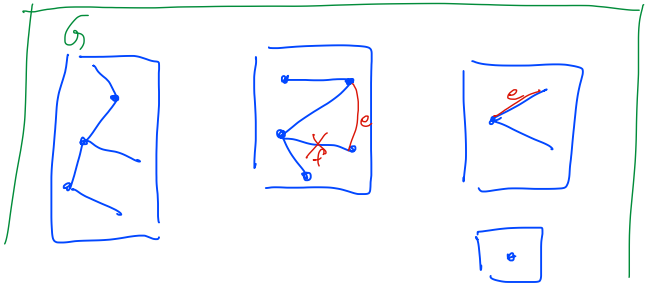


Consecuencia importante

Intercambio

Sea F bosque generador de G.
Para todo $e \in E(G)$, existe $f \in F$ tal que $F - f + e$ es bosque generador de G.

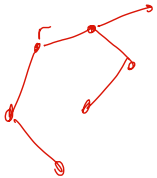
De hecho, si $e \in F$ basta tomar $f = e$
Si $e \notin F$ basta tomar ... $f \in C(F, e) - e$



¿Cómo encontrar algún árbol generador en un grafo conexo?

(o un bosque generador en un grafo cualquiera)

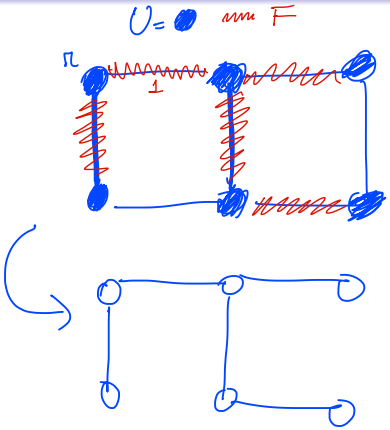
Primera estrategia:
Partir de un vértice r y agregar aristas (sin crear ciclos) hasta que todos los vértices sean alcanzados.



Algoritmos de búsqueda

Algoritmo genérico de búsqueda / crecer un árbol desde un vértice:

Buscar árbol que cubra la CC de un vértice r dado:



Búsqueda genérico:

Entrada: $G = (V, E)$, $r \in V$
$U \leftarrow \{r\}$ // Nodos visitados
$F \leftarrow \emptyset$ // Aristas de solución
mientras $\delta(U) \neq \emptyset$ **hacer**
  Elegir $e = uv \in \delta(U)$, $u \in U$, $v \notin U$
  $U \leftarrow U + v$
  $F \leftarrow F + e$
fin
devolver (U, F)

Correctitud del algoritmo:

G conexo.

Búsqueda genérico:

Entrada: $G = (V, E)$, $r \in V$
$U \leftarrow \{r\}$, // Nodos visitados
$F \leftarrow \emptyset$ // Aristas de solución
mientras $\delta(U) \neq \emptyset$ **hacer**
  Elegir $e = uv \in \delta(U)$, $u \in U$, $v \notin U$
  $U \leftarrow U + v$
  $F \leftarrow F + e$
fin
devolver (U, F)

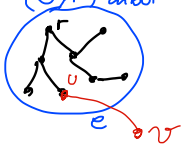
Correctitud $|V| = n$

→ ① Termina. En cada iteración U aumenta en un vértice

② (U, F) es siempre árbol

↳ Al ppio: $(\{r\}, \emptyset)$

↳ Al ppio de cada iteración



$(U + v, F + uv)$ es árbol

- conexo (v es alcanzable desde cualquier $x \in U$)
- acíclico

Ciclo $C \not\ni P + e$

↑ camino de v a v ✗

① Al final $U = V$

Si $U \neq V$ ↝ El alg. termina cuando $\delta(U) = \emptyset$.

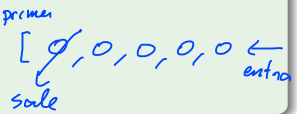
En un grafo conexo $\delta(U) = \emptyset \Rightarrow U = \emptyset$ ó $U = V$.

$\not\emptyset$

Casos especiales: Búsqueda en amplitud y búsqueda en profundidad:

Búsqueda en aplitud (BFS):

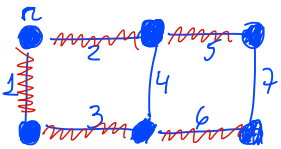
Entrada: $G = (V, E)$, $r \in V$
$U \leftarrow \{r\}$, $F \leftarrow \emptyset$
Cola $\leftarrow \emptyset$.
Insertar aristas de $\delta(r)$ en Cola.
mientras Cola $\neq \emptyset$. **hacer**
  Extraer primer e de Cola.
  si $e \in \delta(U)$, $u \in U$, $v \notin U$
   entonces
    $U \leftarrow U + v$
    $F \leftarrow F + e$
    Insertar aristas de $\delta(v)$ en Cola
  fin
fin
devolver (U, F)



Búsqueda en profundidad (DFS):

Entrada: $G = (V, E)$, $r \in V$
$U \leftarrow \{r\}$, $F \leftarrow \emptyset$
Pila $\leftarrow \emptyset$.
Insertar aristas de $\delta(r)$ en Pila.
mientras Pila $\neq \emptyset$. **hacer**
  Extraer última e de Pila.
  si $e \in \delta(U)$, $u \in U$, $v \notin U$
   entonces
    $U \leftarrow U + v$
    $F \leftarrow F + e$
    Insertar aristas de $\delta(v)$ en Pila
  fin
fin
devolver (U, F)

Ejemplo



BFS

- U

§ F

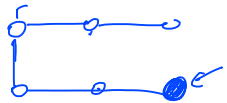
COLA = ~~1~~ ~~2~~ ~~4~~ ~~3~~ ~~2~~ ~~4~~ ~~5~~ ~~3~~ ~~4~~ ~~6~~ ~~5~~ ~~7~~ ~~6~~ ~~7~~

ES(U)

Prop: 1.

4 arista aparece 2 veces.

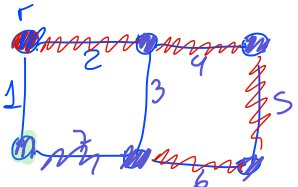
Prop: 2



BFS: Propuesto (calcula distancias)

Ejemplo

DFS



PILA = 1 ~~2~~ ~~2~~ ~~3~~ ~~4~~ ~~4~~ ~~5~~ ~~5~~ ~~6~~ ~~6~~ ~~3~~ ~~7~~