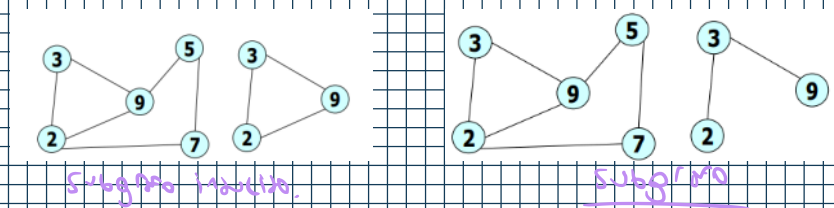


# Grafos resumen

lunes, 19 de junio de 2023 13:53

- Longitud de camino --> aristas intermedias
- Camino simple --> camino en que todos los nodos salvo a veces el primero y el último son diferentes
- Ciclo --> camino en que el 1ro=último
- Bucle --> Ciclo de longitud 1
- Subgrafo inducido --> Tiene que tener sí o sí todas las aristas de los vértices que abarca
- Subgrafo --> lo de arriba no es necesario jajax



- Conexo --> todos los nodos conectados
- Bosque --> grafo sin ciclos
- Árbol libre --> bosque conexo
- Árbol --> árbol libre en el que un nodo es origen
- En grafo  $G$  no dirigido con  $v$  vértices y  $m$  aristas, entonces

$$m \leq (v \cdot (v-1)) / 2 \text{ SIEMPRE}$$

Si  $G$  es conexo y puede tener ciclos,  $m \geq v-1$ . Conexo pero SIN ciclos  $m = v-1$

No conexo y SIN ciclos  $m \leq v-k$  ( $k$  siendo componentes conexas)

Dirigido y sin aristas que vayan a sí mismo  $m \leq v \cdot (v-1)$

Conexo  $\rightarrow m \geq v-1$   
 No  $\rightarrow m = v-1$   
 Bq  $\rightarrow m \leq v-1$

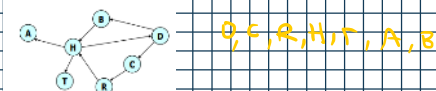
- Fuertemente conexo --> hay un camino desde cualquier vértice hasta cualquier otro
- Debilmente conexo --> Si le sacas el sentido a las flechitas es conexo
- No conexo --> no todos conectados
- Componente fuertemente conexo: partes del grafo que sí son fuertemente conexas

- Matriz de  $V \times V$ , se usa cuando el nro de aristas se acerca a  $v \cdot v$  (denso)  $(O(v^2))$
- Lista guarda peso de arista en el nodo siguiente  $(O(v))$

## Recorridos

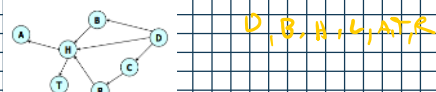
DFS --> orden  $(v+e)$

Recorre un camino hasta que no puede seguir, va para atrás hasta que encuentra otro y repite

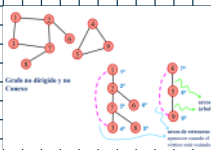


BFS --> orden  $(v+e)$

Es como por niveles, lo tiras como un árbol



## Diagrama de relaciones DFS



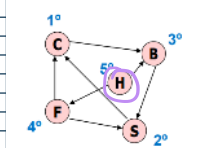
Ir armando como las conexiones de los nodos medio a lo árbol (estirarlos).  
 Una vez que conectaste los nodos (que no se repiten) se agregan las flechas que faltan

- Arco: conducen a vértices no visitados
- Forward: arcos de un antecesor a un descendiente (no hijo)
- Backward: de uno que está después a uno anterior pero no inmediatos.
- Cross: ninguna de las anteriores

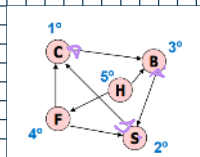
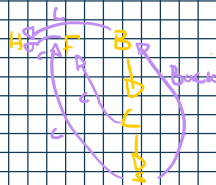
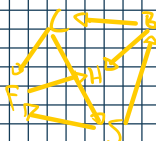
- En no dirigidos, los arcos que van a nodos ya dirigidos se llaman de retroceso (sí, aunque no retrocedan)

$V \rightarrow V + E \rightarrow V$   $\rightarrow$  Componentes fuertemente conexos

Componentes conexas



fuerte



fuerte

fuerte

- Apilar o numerar los vértices en postorden.
- Dibujar el grafo invirtiendo las flechitas
- Aplicarle dfs (hacer bosque de expansión) arrancando desde el tope de la pila (o último numerito)

sort topología

Con arreglo  $\rightarrow$  orden(v)

1	2	3	4	5	6	7
---	---	---	---	---	---	---

proceso  
de entrada  
(ordenar)  
(ordenar)

- Seleccionar un nodo con grado 0, printearlo y bajarle el grado a los adyacentes
- Repetir hasta visitar todos los v

Con arreglo y pila/cola  $\rightarrow$  orden (v+a)

- Casi igual al anterior pero apilando los que tienen orden 0

DFS  $\rightarrow$  orden (v+a)

- Se apilan los vértices en post orden, se lista a medida que se desapila.



sort: 02, 04, 03, 05

camino minto

Grafos sin pesos

- Se hace un cuadrito. Arrancan todos en infinito salvo el inicio.
- Se toma el que tenga el valor más chico, se lo marca como conocido, se suma la cantidad de aristas por las que se pasó +1 y se pone ese valor a los vértices adyacentes no conocidos

Vértice	Distancia	Anterior	Conocido
1	infinito		
2	infinito		
3	0		
4	infinito		

Dijkstra  $\rightarrow$  Solo workea para pesos positivos. Orden v

- Se toma el que tenga el valor más chico, se lo marca como conocido, se fija si la suma de lo que cuesta llegar al nodo en el que estás + el peso de la arista siguiente es < que el valor que ya tenía. Si es más chico, se actualiza eso y se escribe el anterior

Vértice	Distancia	Anterior	Conocido
1	infinito		
2	infinito		
3	0		
4	infinito		

Grafos con pesos positivos y negativos

- Se encola el origen
- Mientras no se vacíe la cola, desencolo.
- Para cada adyacente, si tengo que actualizar actualizo
- Solo si actualicé y si no está en la cola, encolo lo que actualicé



ver	Peso	Ant	conocido
s	0		
u	-1	v	

- Mientras no se vacíe la cola, desencolo.
- Para cada adyacente, si tengo que actualizar actualizo
- Solo si actualicé y si no está en la cola, encolo lo que actualicé



ver	Peso	Ant	conocido
s	0		
u	5	v	
v	2	s	

### Grafos acíclicos -> orden v+e

- Se ordenan topológicamente (se hace vector)
- Se hace una tablita de dijkstra
- Se van apilando los vértices con orden 0: cuando se desapila una se baja el orden de sus adyacentes y se actualiza en caso de que cumpla la regla (SUMANDO)



Vertice	Peso	Anterior	conocido
1	i		1
2	0		1
3	51	4	1
4	1	2	1
5	53	3	1
6	57	5	1
7	59	6	1

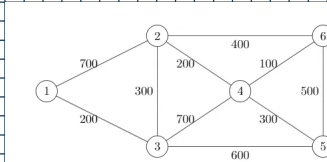
1	2	3	4	5	6	7
0	0	0	0	0	0	0

C \*\* \*\* \*\* \*\*  
 \*\* \*\* \*\*

### Entre todos los pares de vértices (Floyd) -> orden $V^3$

Se hace una matriz de  $V \times V$ . Se completa con los valores de las aristas, infinito en las que no existen. Se selecciona fila 1 columna 1 y se comparan TODOS los elementos de la tabla con la suma de los elementos de la intersección correspondiente. Si da un valor menor, se pone ese valor.

Este método da el camino mínimo entre TODOS los vértices

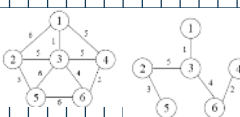


	1	2	3	4	5	6
1	0	700	200	i	i	i
2	700	0	300	200	i	400
3	200	300	0	700	600	i
4	i	200	700	0	300	100
5	i	i	600	300	0	500
6	i	400	i	100	500	0

	1	2	3	4	5	6
1	0	500	200	700	800	800
2	500	0	300	200	500	300
3	200	300	0	500	600	600
4	700	200	500	0	300	100
5	800	500	600	300	0	400
6	800	300	600	100	400	0

### Árbol de expansión mínima

- Árbol con las aristas del grafo que conectan todos los vértices con el mínimo costo



### Algoritmo de prim -> orden $|V|^2$

- Se hace una tablita parecida a dijkstra. Se toma el vértice con menos peso, se lo conecta en el árbol (conectado desde la arista anterior) y se lo marca como visitado. Se actualiza si es menor. NO SE SUMAN

### Algoritmo de kruskal -> orden $|V|^2$

- Ordenar todas las aristas en orden creciente según el peso que tengan
- Dibujar los vértices del nodo sin las aristas (los vértices flotando)
- Si no genera un ciclo, agregar la arista

