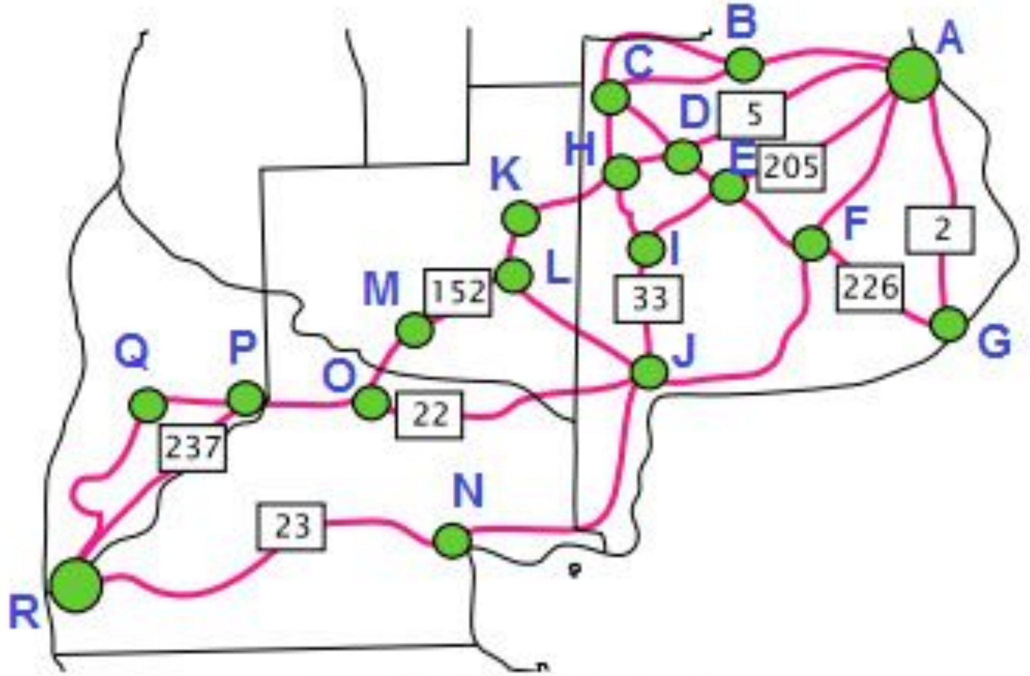
Grafos

1. **Ejemplos y terminología**
2. Representaciones
3. Recorridos

Grafos

Ejemplo 1: Mapa de ciudades

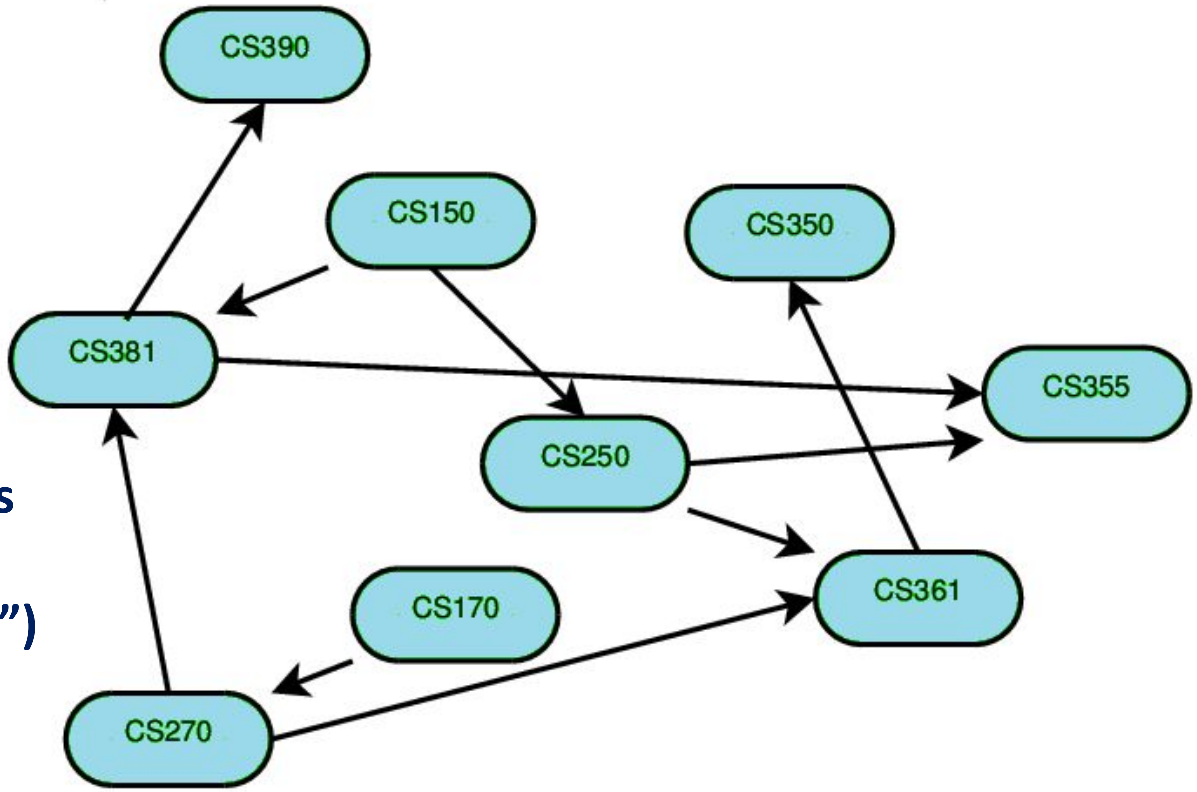


Rutas entre ciudades

Ciudades conectadas por *Rutas*

Grafos

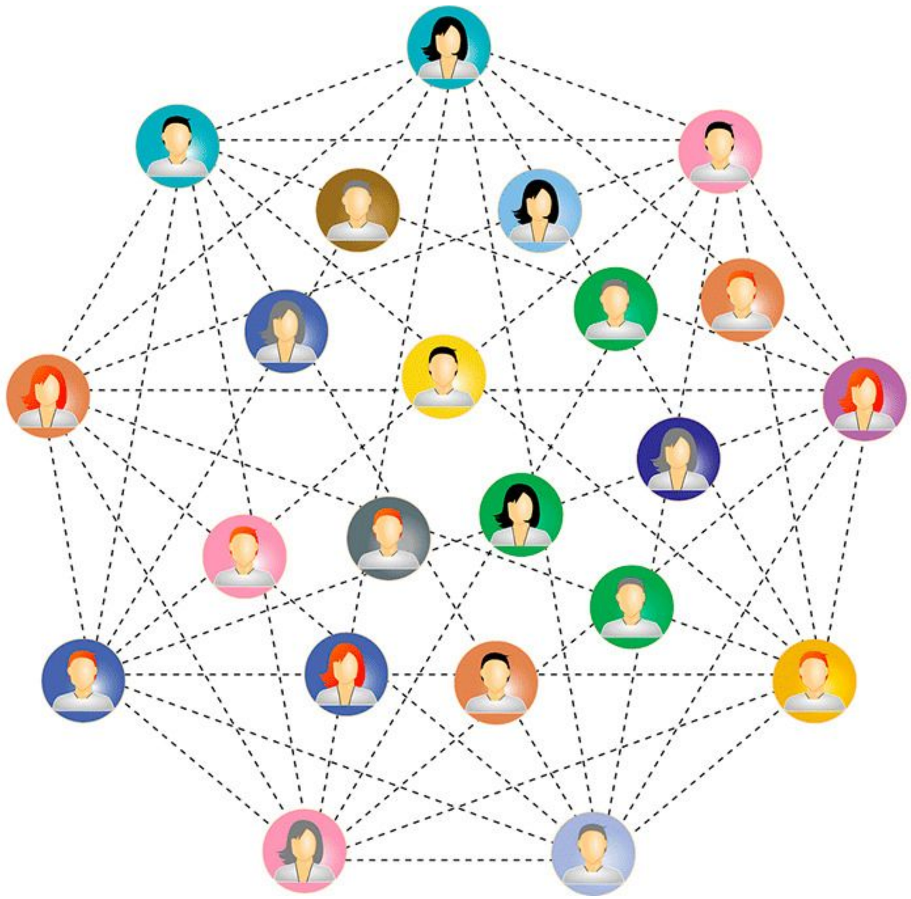
Ejemplo 2: Pre-requisitos de un curso



Cursos conectados por sus correlativas
(*relación* de “prerrequisito”)

Grafos

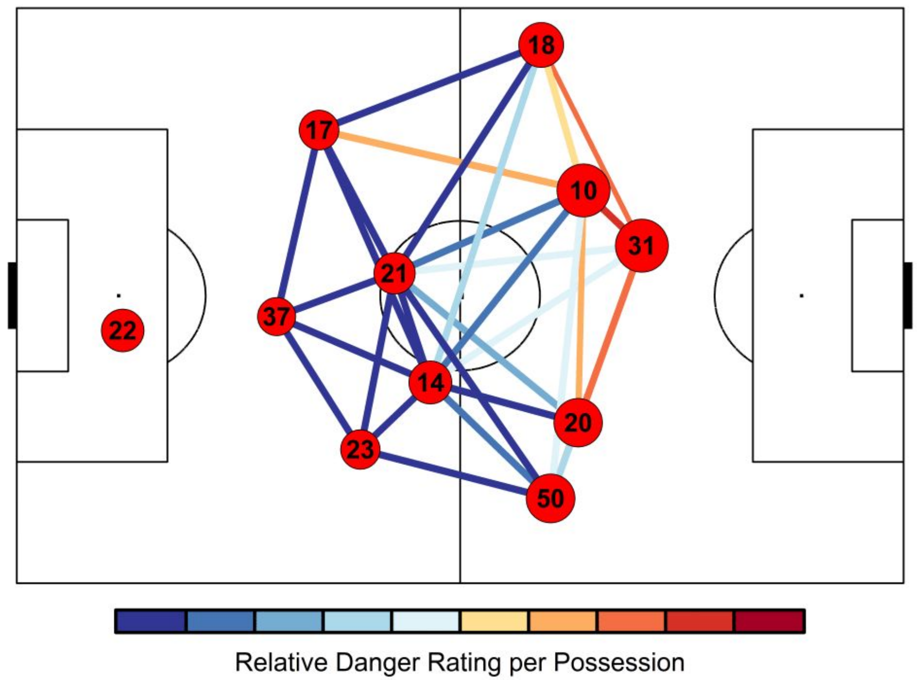
Ejemplo 3: Redes sociales



Personas conectadas en una red social

Grafos

Ejemplo 4: red de pases de un partido de fútbol



Red de pases para el Barcelona y el AC Milan de un partido de Liga de Campeones. Las flechas más oscuras y gruesas indican más pases entre cada jugador.

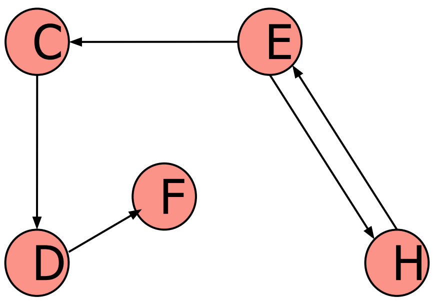
Grafos
Terminología (1 de 9)

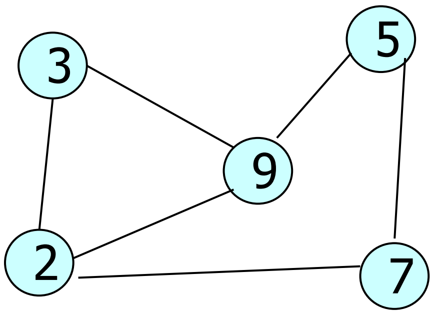
- ***Grafo*** → *es una estructura de datos no lineal que consiste de un conjunto finito de vértices y un conjunto finito de aristas.*
- ***Grafo****: Un grafo es un par de conjuntos (V,E), donde V es un conjunto de vértices o nodos y E es un conjunto de pares (u,v), u,v ∈ V, llamados aristas o arcos.*
- ***Grafo dirigido****: la relación sobre V no es simétrica. Arista ≡ par ordenado (u,v).*
- ***Grafo no dirigido****: la relación sobre V es simétrica. Arista ≡ par no ordenado {u,v}, u,v ∈ V y u ≠ v.*

Grafos

Terminología (2 de 9)

Ejemplos





Grafo dirigido *G(V,E)*

V = {C,D,E,F,H}

E={(C,D),(D,F),(E,C),(E,H), (H,E)}

Grafo no dirigido *G(V,E)*

V = {2,3,5,7,9}

E={{2,3},{2,7},{2,9},{3,9},{5,7},{5,9}}

*Dado un grafo G(V,E), v es **adyacente** a u si existe una arista (u,v) Є E:*

- *en un grafo dirigido, (u,v) Є E, parte de u e incide en v,   u→ v*
- *en un grafo no dirigido, {u,v} Є E, incide en los nodos u, v.*

Grafos
Terminología (3 de 9)

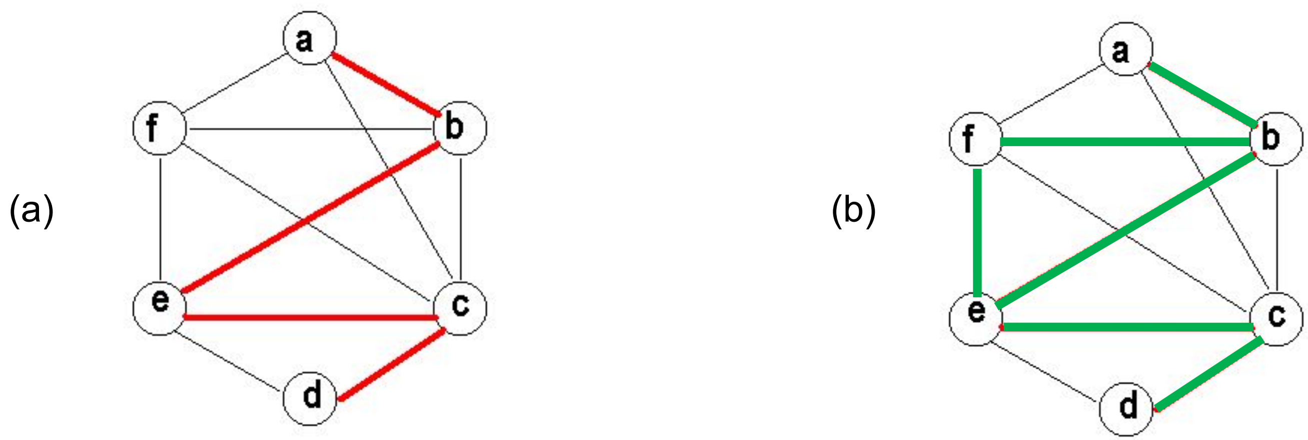
Grado

- *En grafos no dirigidos:*
  - *El **grado** de un nodo: número de arcos que inciden en él; es el número de vértices adyacentes*
- *En grafos dirigidos:*
  - *existen el grado de salida (**outdegree**) y el grado de entrada (indegree).*
    - *el **outdegree** es el número de arcos que parten de él y*
    - *el **indegree** es el número de arcos que inciden en él.*
  - *El **grado** del vértice será la suma de los grados de entrada y de salida.*
- ***Grado de un grafo**: máximo grado de sus vértices.*

Grafos
Terminología (4 de 9)

- ***Camino*** *desde u Є V a v Є V : secuencia* $v_1, v_2, \ldots, v_k$ *tal que* $u=v_1$, $v=v_k$, *y* (v_{i-1},v_i) Є *E, para* $i = 2, \ldots ,k$.

  *Camino desde **a** a **d** → <a,b,e,c,d>* *camino desde **a** a **d** → <a,b,e,f,b,e,c,d>*



- ***Longitud de un camino****: número de arcos del camino.*

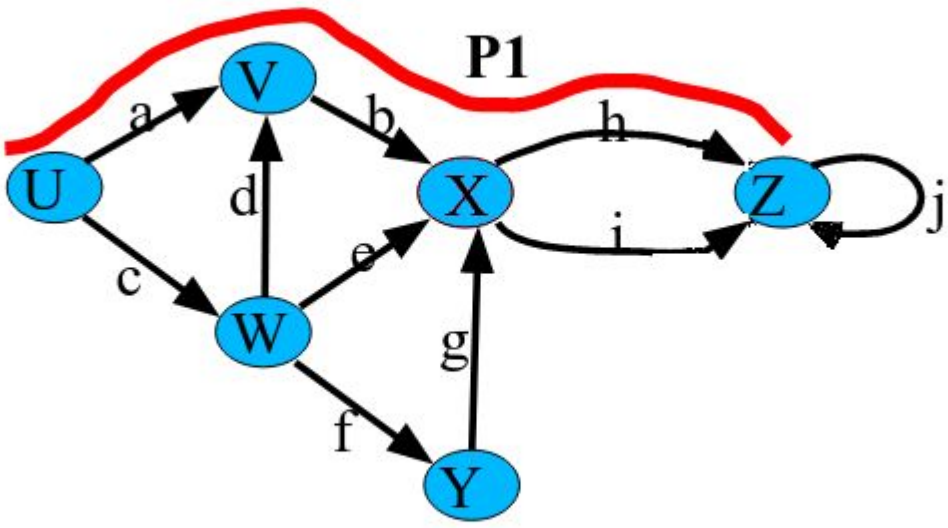
  *Ejemplos: long. del camino (a) desde **a** a **d** → 4*

  *long. del camino (b) desde **a** a **d** → 7*

Grafos

Terminología (5 de 9)

Camino simple: *camino en el que todos sus vértices son distintos -excepto para caminos que son ciclo, donde el primero y el último son iguales-. P1 es un camino simple desde U a Z.*

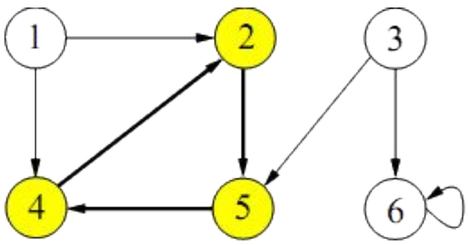


Ejemplos anteriores: (a) es camino simple , (b) no lo es.

Grafos
Terminología (6 de 9)

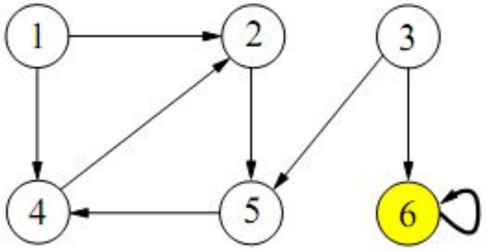
- ***Ciclo***: *camino desde* $v_1, v_2, \ldots, v_k$ *tal que* $v_1 = v_k$

  Ej: <2,5,4,2> es un ciclo de longitud 3.

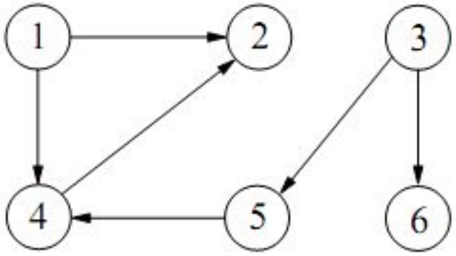


  El ciclo es simple si el camino es simple.

- ***Bucle***: *ciclo de longitud 1.*



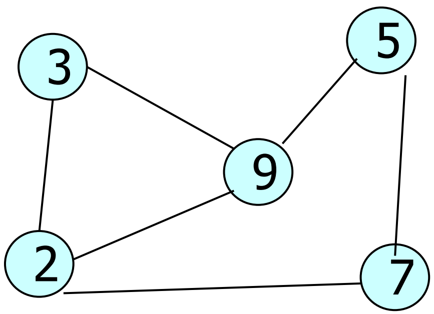
- ***Grafo acíclico***: *grafo sin ciclos.*



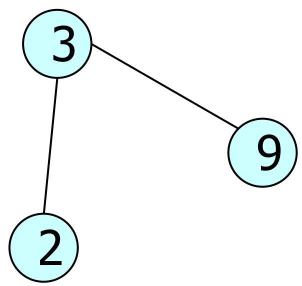
Grafos

Terminología (7 de 9)

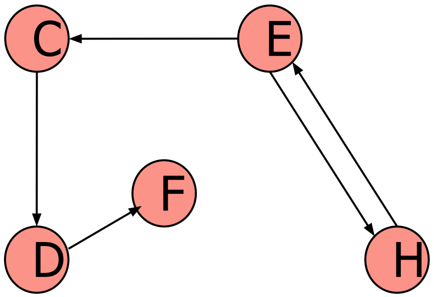
*Dado un grafo G=(V, E), se dice que G´=(V´, E´) es un **subgrafo** de G, si* $V' \subseteq V$ *y* $E' \subseteq E$.



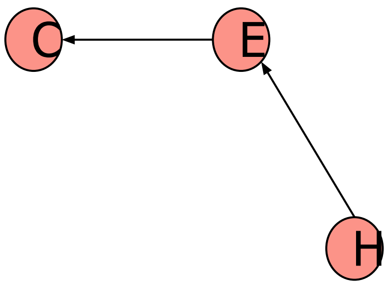
G = (V, E)



G´ = (V´, E´)



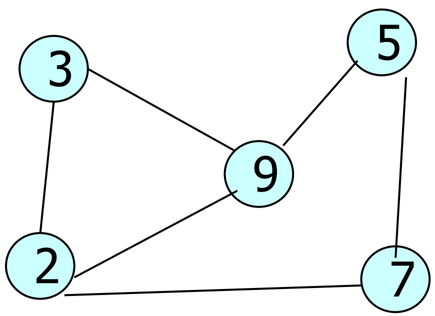
G = (V, E)



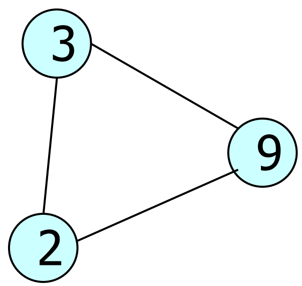
G´ = (V´, E´)

Grafos
Terminología (8 de 9)

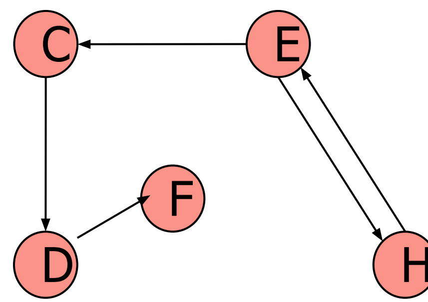
*Un **subgrafo inducido** por* $V' \subseteq V : G' = (V', E')$ *tal que* $E' = \{(u,v) \in E \mid u,v \in V'\}$.



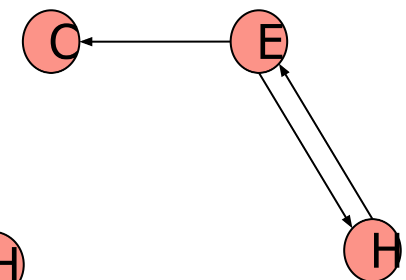
G = (V, E)



G´ = (V´, E´)



G = (V, E)

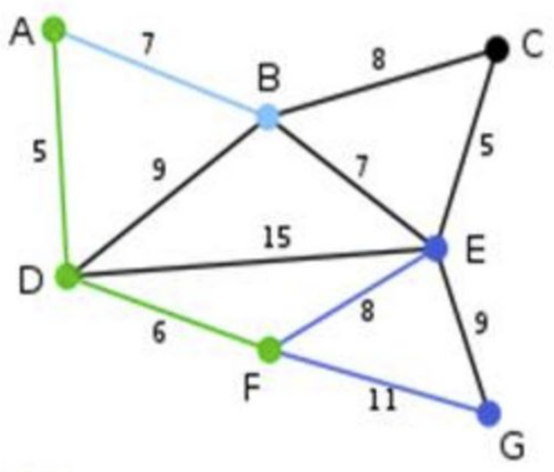


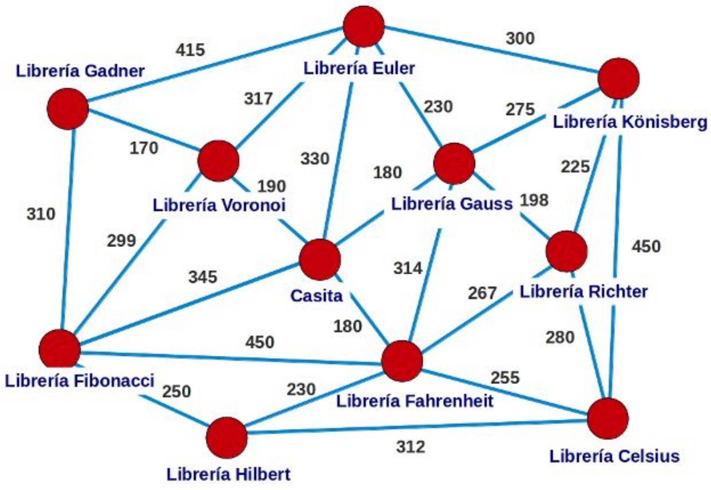
G´ = (V´, E´)

Dado subconjunto de vértices S'⊆S, el subgrafo inducido por S', es el grafo que tiene como conjunto de vértices a S' y como conjunto de aristas a todas las aristas de G cuyos dos extremos están en S'

Grafos
Terminología (9 de 9)

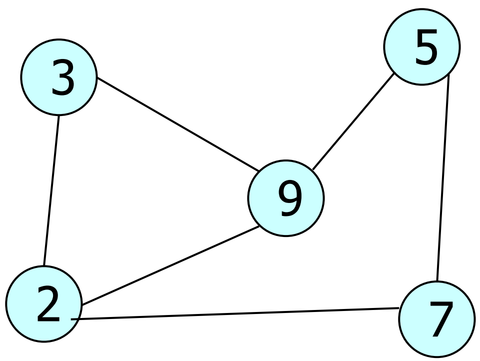
- *En un grafo* ***ponderado, pesado o con costos****, cada arco o arista tiene asociado un valor o etiqueta*



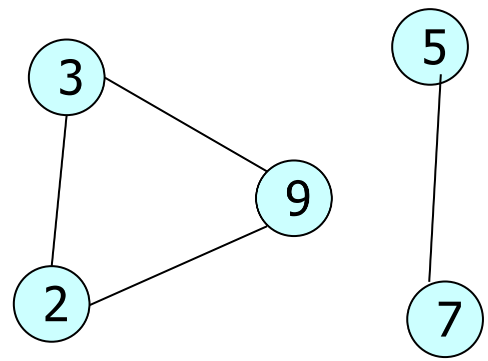


Grafos no dirigidos
Conectividad (1 de 3)

- *Un grafo no dirigido es **conexo** si hay un camino entre cada par de vértices.*



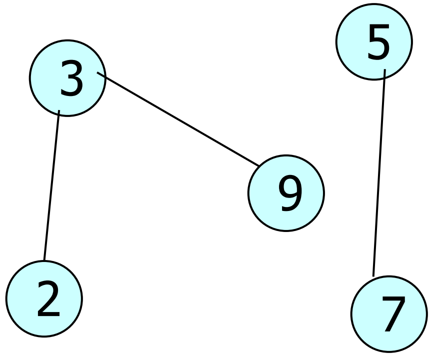
Conexo



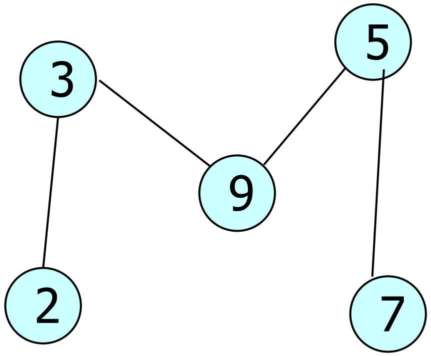
No Conexo

Grafos no dirigidos
Conectividad (2 de 3)

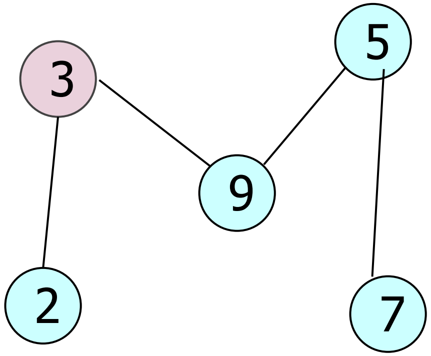
- *Un **bosque** es un grafo sin ciclos.*
- *Un **árbol libre** es un grafo sin ciclo y conexo.*
- *Un **árbol** es un grafo conexo y sin ciclo en el que un nodo se ha designado como raíz.*



Bosque



Árbol libre



Árbol

Grafos no dirigidos
Conectividad (2 de 3)

Sea G un grafo ***no dirigido*** *con* ***n*** *vértices y* ***m*** *arcos, entonces:*

$$\sum_{v \in G} grado(v) = 2*m$$

y siempre: $m \leq (n*(n-1))/2$

Si G es un árbol: $m=n-1$

Si G es conexo: $m \geq n-1$

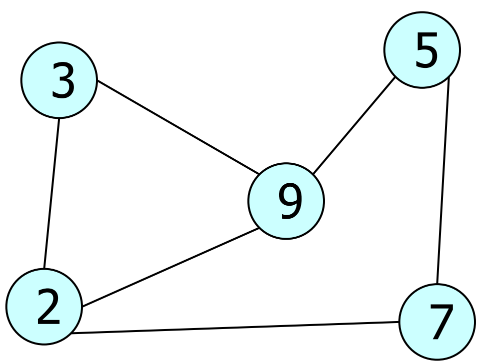
Si G es un bosque: $m \leq n-1$

Grafos no dirigidos

Componentes conexas

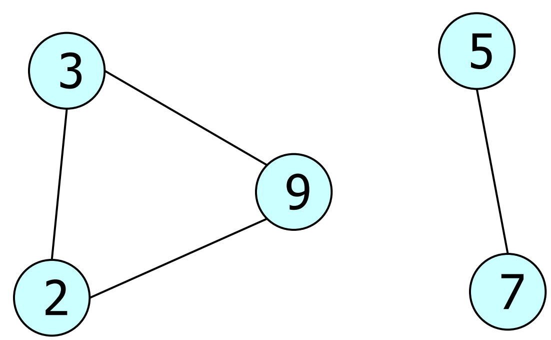
*En un grafo no dirigido, una **componente conexa es un** subgrafo conexo tal que no existe otra componente conexa que lo contenga. Es un **subgrafo conexo maximal**.*

*Un grafo no dirigido es **no conexo** si está formado por varias componentes conexas.*



Conexo

No Conexo

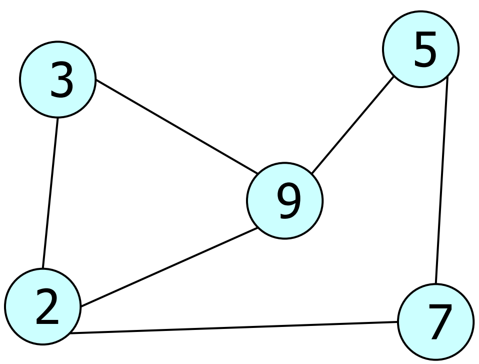


Grafos no dirigidos

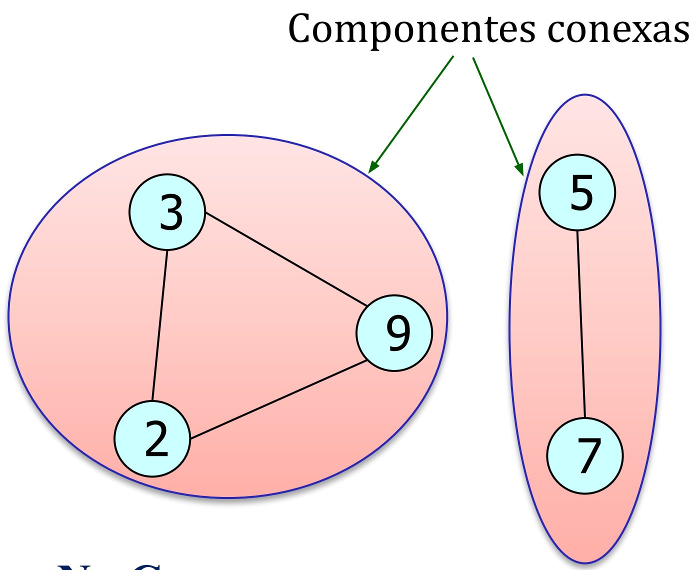
Componentes conexas

*En un grafo no dirigido, una **componente conexa es un** subgrafo conexo tal que no existe otra componente conexa que lo contenga. Es un **subgrafo conexo maximal**.*

*Un grafo no dirigido es **no conexo** si está formado por varias componentes conexas.*



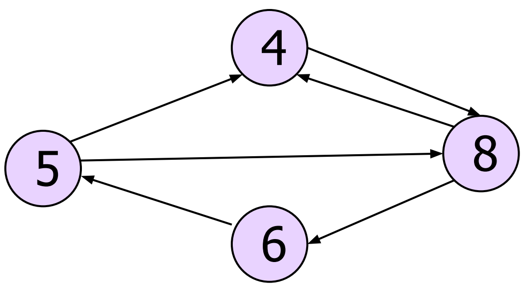
Conexo



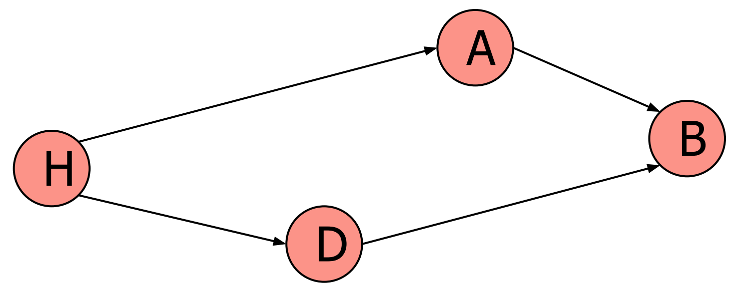
No Conexo

Grafos dirigidos
Conectividad

- *v es **alcanzable desde** u, si existe un camino de u a v.*
- *Un grafo dirigido se denomina **fuertemente conexo** si existe un camino desde cualquier vértice a cualquier otro vértice* del grafo.



Fuertemente Conexo



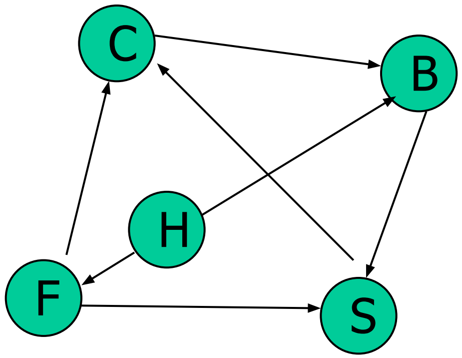
No Fuertemente Conexo
Débilmente Conexo

- *Si un grafo dirigido no es fuertemente conexo, pero el grafo subyacente (sin sentido en los arcos) es conexo, el grafo es **débilmente conexo**.*

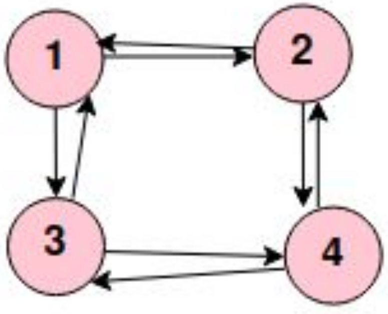
Grafos dirigidos
Componentes Conexas

En un grafo dirigido, una ***componente fuertemente conexa****, es el máximo subgrafo fuertemente conexo.*

Un grafo dirigido es ***no fuertemente conexo*** *si está formado por varias componentes fuertemente conexas.*



No Fuertemente Conexo

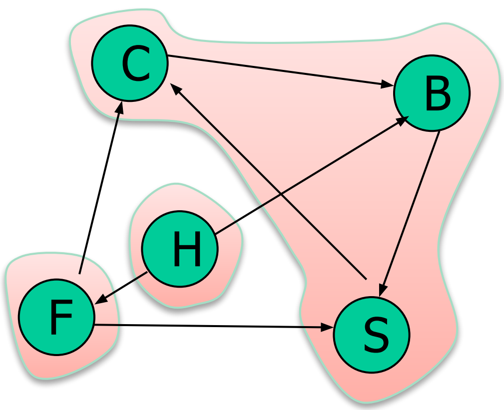


Fuertemente Conexo

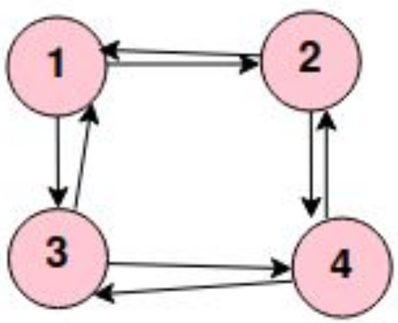
Grafos dirigidos
Componentes Conexas

*En un grafo dirigido, una **componente fuertemente conexa**, es el máximo subgrafo fuertemente conexo.*

*Un grafo dirigido es **no fuertemente conexo** si está formado por varias componentes fuertemente conexas.*



No Fuertemente Conexo



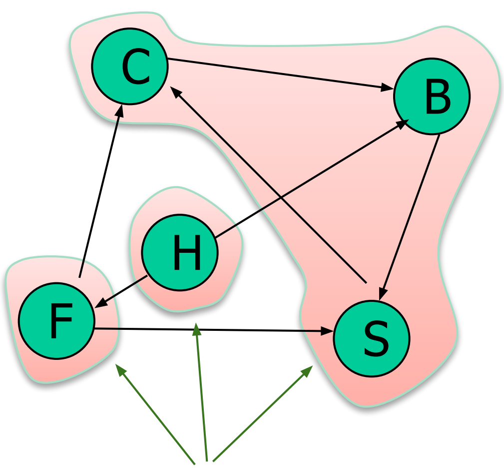
Fuertemente Conexo

Grafos dirigidos
Componentes Conexas

*En un grafo dirigido, una **componente fuertemente conexa**, es el máximo subgrafo fuertemente conexo.*

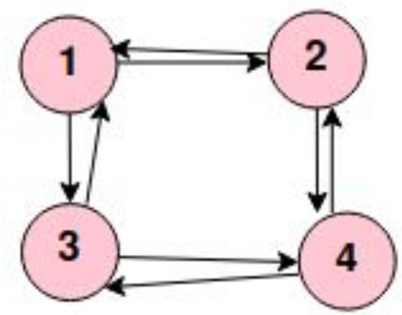
*Un grafo dirigido es **no fuertemente conexo** si está formado por varias componentes fuertemente conexas.*

No Fuertemente Conexo



Componentes fuertemente conexas

Fuertemente Conexo



Grafos

1. Ejemplos y terminología
2. **Representaciones**
3. Recorridos

Agenda - Grafos

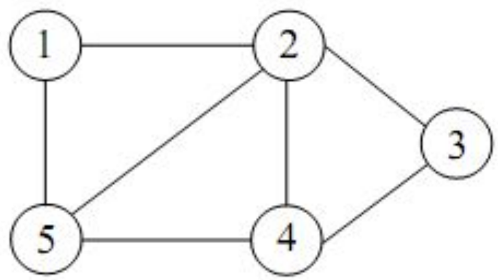
- Representaciones
  - Matriz de Adyacencias
  - Lista de Adyacencias

Representaciones

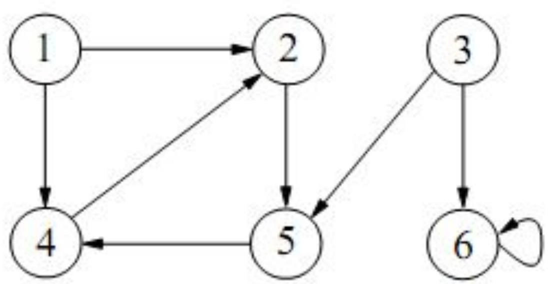
Matriz de Adyacencias

- *G =(V ,E): matriz A de dimensión $|V| \times |V|$.*
- *Valor a_{ij} de la matriz:*

$$a_{ij} = \begin{cases} 1 & \text{si } (i,j) \in E \\ 0 & \text{en cualquier otro caso} \end{cases}$$



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0



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

Representaciones
Matriz de Adyacencias

- ***Costo espacial:*** $O\ (|V|^2)$

- *Representación es útil para grafos con un número de vértices pequeño, o grafos densos* $(|E| \approx |V| \times |V|)$

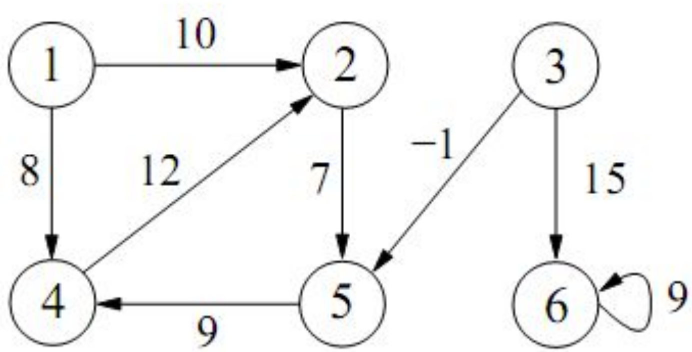
- *Comprobar si una arista (u,v) pertenece a E* $\rightarrow$ *consultar posición A(u,v)*
  Costo de tiempo: $T(|V|,|E|) = O(1)$

Representaciones

Matriz de Adyacencias

- *Representación aplicada a Grafos pesados*
- *El peso de (i,j) se almacena en A (i, j)*

$$a_{ij} = \begin{cases} w(i,j) & \text{si } (i,j) \in E \\ 0 \; o \; \infty & \text{en cualquier otro caso} \end{cases}$$



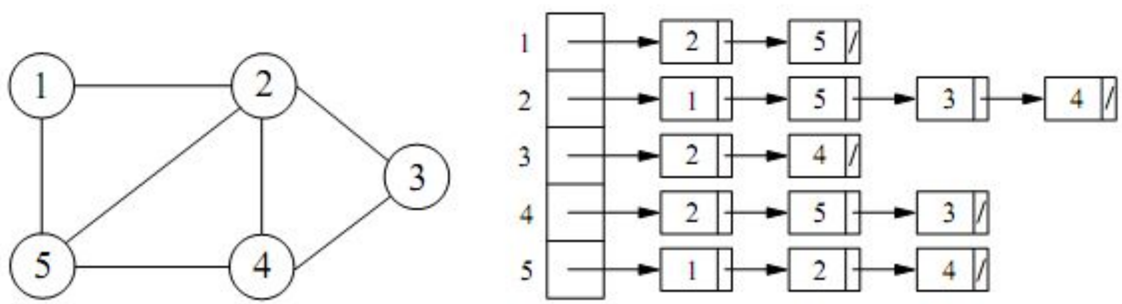
	1	2	3	4	5	6
1	0	10	0	8	0	0
2	0	0	0	0	7	0
3	0	0	0	0	−1	15
4	0	12	0	0	0	0
5	0	0	0	9	0	0
6	0	0	0	0	0	9

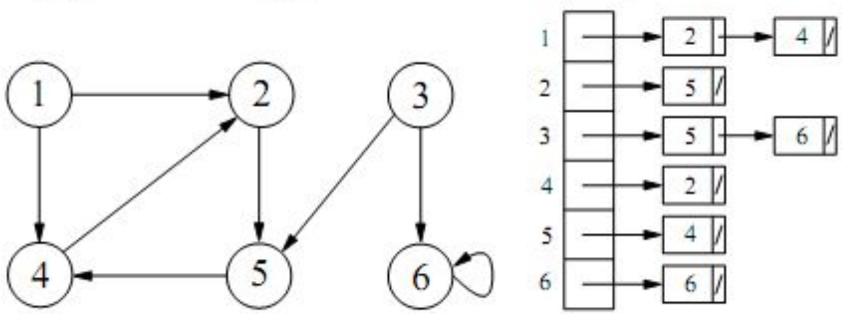
Representaciones
Lista de Adyacencias

G=(V,E): vector de tamaño |V|

*En la posición **i** de V → referencia a una lista enlazada de elementos (lista de adyacencia)*

*Los elementos de la lista representan los vértices adyacentes al vértice que está en **i***





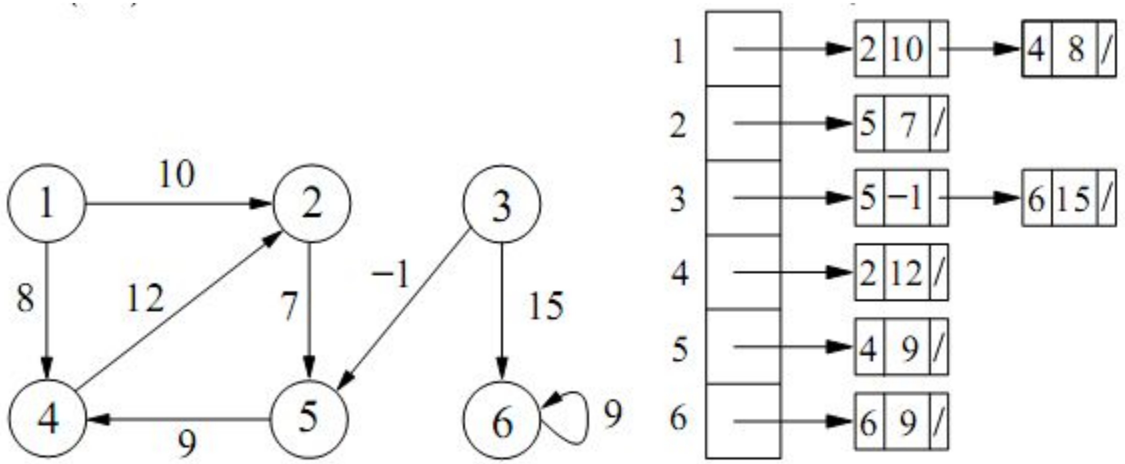
Representaciones
Lista de Adyacencias

- *Si G es dirigido, la suma de las longitudes de las listas de adyacencia será* $|E|$
- *Si G es no dirigido, la suma de las longitudes de las listas de adyacencia será* $2|E|$.
- ***Costo espacial:*** $O\ (|V|+|E|)$ *-sea dirigido o no-*
- *Representación apropiada para grafos con* $|E|$ *menor que* $|V|^2$.
- *Desventaja: si se quiere comprobar si una arista (u,v) pertenece a E* $\Rightarrow$ *buscar* ***v*** *en la lista de adyacencia de u.*

Costo de tiempo: $T\ (|V|,|E|)$ *será* O *(Grado G)* $\subseteq O\ (|V|)$

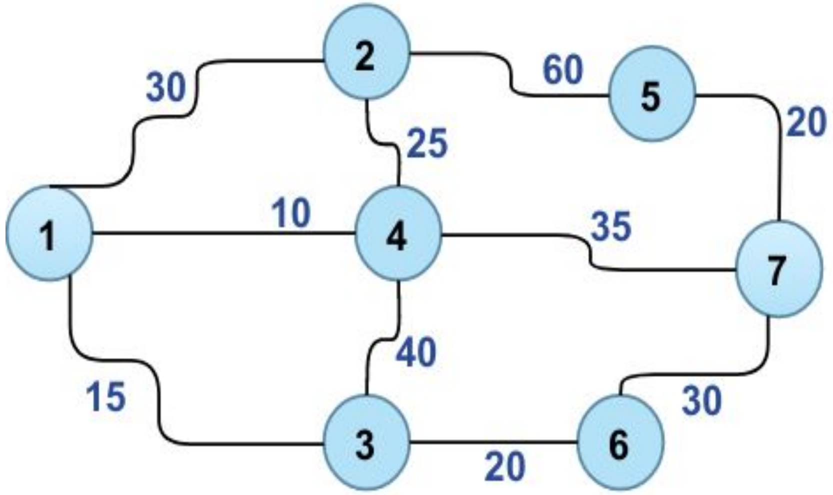
Representaciones
Lista de Adyacencias

- *Representación aplicada a Grafos pesados*
- *El* ***peso de (u,v)*** *se almacena en la lista de adyacencia de* ***u****.*



Problema: El Guía de Turismo

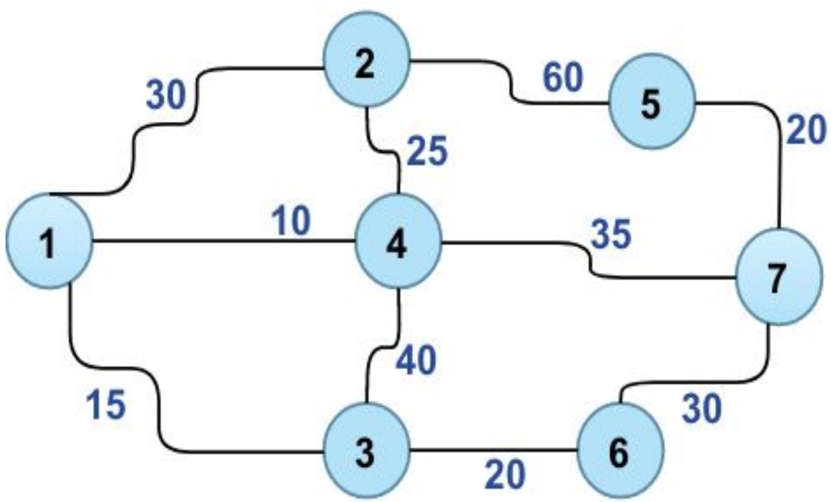
El señor H es un guía de turismo de la ciudad de Buenos Aires. Su trabajo consiste en mostrar a grupos de turistas diferentes **puntos de interés** de la ciudad.
Estos puntos de interés están **conectados por rutas en ambos sentidos**. Dos puntos de interés vecinos tienen un servicio de bus que los conecta, con una limitación en el **número máximo de pasajeros** que puede transportar. No es siempre posible para el señor H transportar de una única vez a todos los turistas a un destino en particular.



Por ejemplo, consideremos el siguiente mapa con **7 puntos de interés**, donde las aristas representan las rutas y **el peso de ellas representa el límite máximo de pasajeros a transportar por el servicio de bus**. Su misión es indicarle al Sr. H cuál es el menor número de viajes que deberá realizar para llevar al grupo de turistas de un origen a un destino.

Problema: El Guía de Turismo 

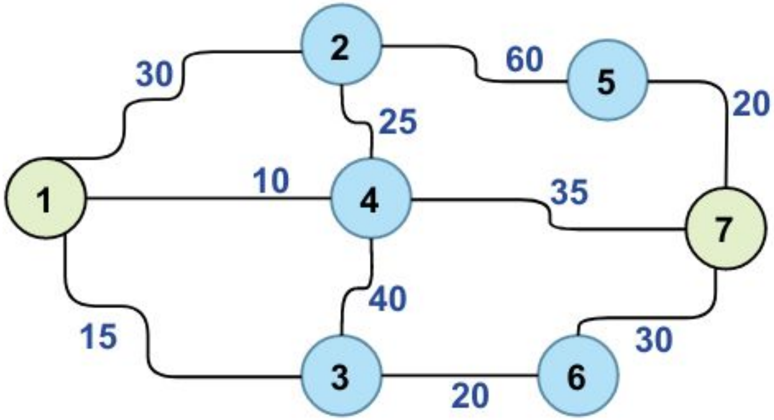
El señor H es un guía de turismo de la ciudad de Buenos Aires. Su trabajo consiste en mostrar a grupos de turistas diferentes **puntos de interés** de la ciudad.
Estos puntos de interés están **conectados por rutas en ambos sentidos**. Dos puntos de interés vecinos tienen un servicio de bus que los conecta, con una limitación en el **número máximo de pasajeros** que puede transportar. No es siempre posible para el señor H transportar de una única vez a todos los turistas a un destino en particular.



Por ejemplo, consideremos el siguiente mapa con **7 puntos de interés**, donde las aristas representan las rutas y **el peso de ellas representa el límite máximo de pasajeros (capacidad del ómnibus).** Su misión es indicarle al Sr. H qué camino debe tomar para llevar un grupo de turistas de un origen a un destino y haciendo el menor número de viajes.

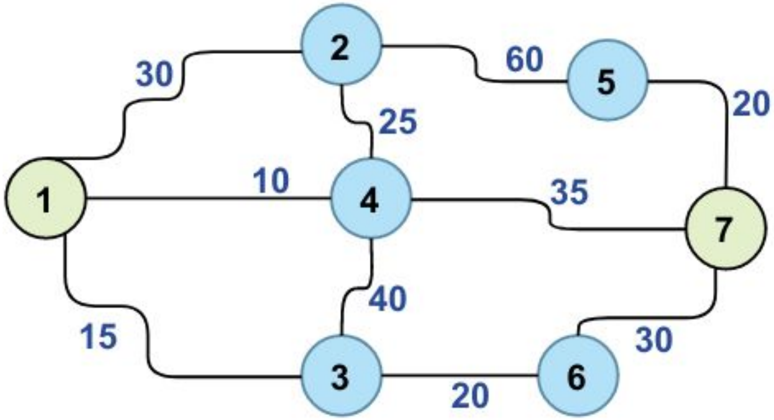
Supongamos que el señor H debe transportar a **99 turistas** del punto **1** al punto **7**.
Cuáles son los recorridos posibles? Cuál implica realizar el menor número de viajes?

Problema: El Guía de Turismo



Puntos de interés	Cant turistas x viaje	Cant. viajes
1 2 4 3 6 7	20	6
1 2 4 7	25	5
1 2 5 7	20	6
1 3 4 2 5 7	15	8
1 3 4 7	15	8
1 3 6 7	15	8
1 4 2 5 7	10	11
1 4 3 6 7	10	11
1 4 7	10	11

Problema: El Guía de Turismo 



Puntos de interés	Cant turistas x viaje	Cant. viajes
1 2 4 3 6 7	20	6
1 2 4 7	**25**	**5**
1 2 5 7	20	6
1 3 4 2 5 7	15	8
1 3 4 7	15	8
1 3 6 7	15	8
1 4 2 5 7	10	11
1 4 3 6 7	10	11
1 4 7	10	11

Entonces, para transportar a los **99 turistas** del punto 1 al punto 7, se necesitarán 5 viajes, eligiendo la ruta: **1 → 2 → 4 → 7**

Grafos

Recorrido en profundidad: DFS

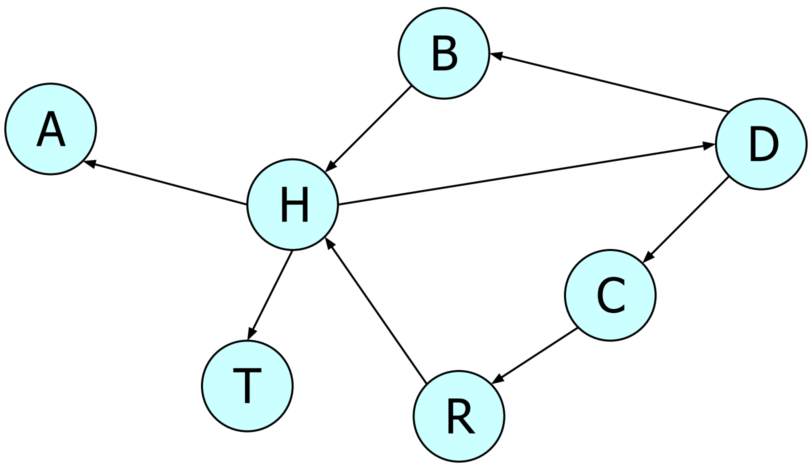
Generalización del recorrido preorden de un árbol

***Estrategia**:*

- *Partir de un vértice determinado **v**.*
- *Cuando se visita un nuevo vértice, explorar cada camino que salga de él*
- *Hasta que no se haya finalizado de explorar uno de los caminos no se comienza con el siguiente*
- *Un camino deja de explorarse cuando se llega a un vértice ya visitado -o sin adyacentes-.*
- *Si existían vértices no alcanzables desde **v** el recorrido queda incompleto; entonces, se debe seleccionar algún vértice como nuevo vértice de partida, y repetir el proceso.*

Grafos

Recorrido en profundidad: DFS



Si tomamos como vértice de partida a D

Se Muestra:

D C R H T A B

Grafos

Recorrido en profundidad: DFS

Esquema recursivo*: dado G = (V , E)*

1. *Marcar todos los vértices como no visitados.*
2. *Elegir vértice **u** como punto de partida.*
3. *Marcar **u** como visitado.*
4. *∀ **v** adyacente a **u**,**(u,v)** ∈ E, si **v** no ha sido visitado, repetir recursivamente (3) y (4) para **v**.*

- *Finalizar cuando se hayan visitado todos los nodos alcanzables desde u.*
- *Si desde **u** no fueran alcanzables todos los nodos del grafo: volver a (2), elegir un nuevo vértice de partida **v** no visitado, y repetir el proceso hasta que se hayan recorrido todos los vértices.*

Recorrido en profundidad: DFS

```
dfs (grafo)
    inicializar marca en false (arreglo de booleanos);
    para cada vértice v del grafo
            si v no está visitado
            dfs(v);

dfs (v: vértice)
    marca[v]:= visitado;
    para cada nodo w adyacente a v
            si w no está visitado
            dfs(w);
```

Recorrido DFS: Tiempo de ejecución

- *El método **dfs(v)** se aplica únicamente sobre vértices no visitados* $\rightarrow$ *sólo una vez sobre cada vértice.*

- ***dfs(v)** depende del número de vértices adyacentes que tenga (longitud de la lista de adyacencia).*

  $\rightarrow$ *el tiempo de todas las llamadas a **dfs(v)**:* $O(|E|)$

- *añadir el tiempo asociado al bucle de main_dfs(grafo):* $O(|V|)$)

  $\Rightarrow$ *Tiempo del recorrido en profundidad es* $O(|V|+|E|)$.

Recorrido en amplitud: BFS

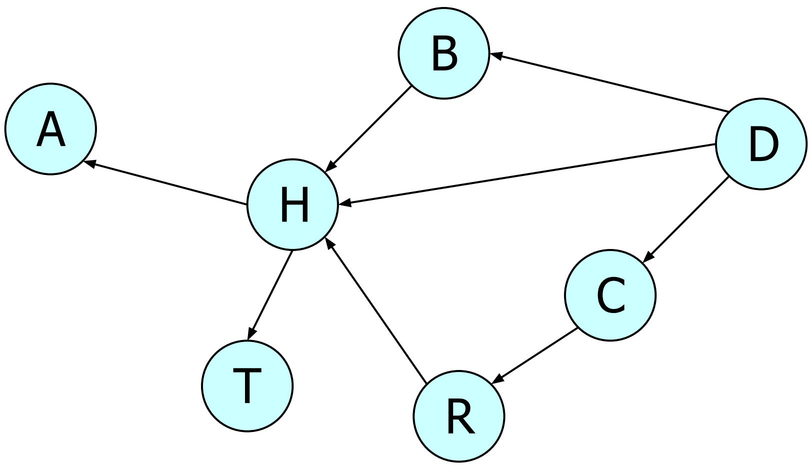
→ *Generalización del recorrido por niveles de un árbol.*

Estrategia*:*

- *Partir de algún vértice **u**, visitar **u** y, después, visitar cada uno de los vértices adyacentes a **u**.*

- *Repetir el proceso para cada nodo adyacente a **u**, siguiendo el orden en que fueron visitados.*

Recorrido en amplitud: BFS



Si tomamos como vértice de partida a D

Se Muestra:

D C H B R T A

Cola:

D C H B R T A

Recorrido en amplitud: BFS

***Esquema iterativo**: dado G = (V , E)*

1. *Encolar el vértice origen u.*
2. *Marcar el vértice u como visitado.*
3. *Desencolar u de la cola*
4. *$\forall$ adyacente a u,(u,v) Є E,*

   si v no ha sido visitado encolar y visitar v

*Si desde **u** no fueran alcanzables todos los nodos del grafo: volver a (1), elegir un nuevo vértice de partida no visitado, y repetir el proceso hasta que se hayan recorrido todos los vértices*

Costo $T(|V|,|E|)$ es de $O(|V|+|E|)$