Grafos

Temario

- Definición
- Descripción y terminología
- Ejemplos
- Representaciones
- Recorridos
- Ordenación topologica

Conceptos Generales

Definición

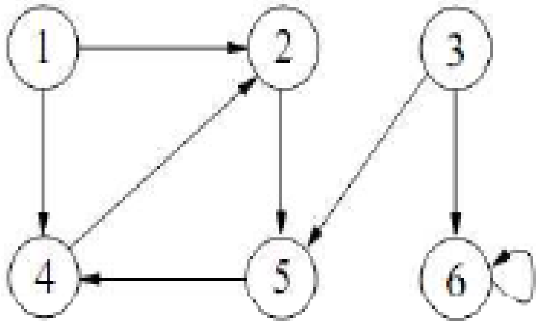
Grafo: (V ,E), V es un conjunto de vértices o nodos, con una relación entre ellos; E es un conjunto de pares (u,v), u,v Є V , llamados aristas o arcos.

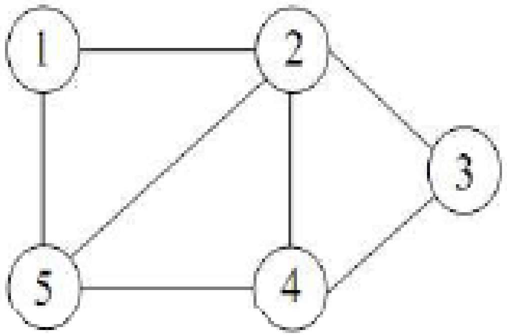
Descripción y terminología

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 Є V y u ≠ v

Descripción y terminología





Ejemplo 1:

Grafo dirigido G(V , E).

V = {1,2,3,4,5,6}

E={(1,2),(1,4),(2,5),(3,5),(3,6),(4,2),
(5,4),(6,6)}

Ejemplo 2:

Grafo no dirigido G(V , E).

V = {1,2,3,4,5}

E= {{1,2},{1,5},{2,3},{2,4},{2,5},
{3,4},{4,5}}

Descripción y terminología

Camino desde u Є V a v Є V : secuencia v1, v2, ... , vk tal que:

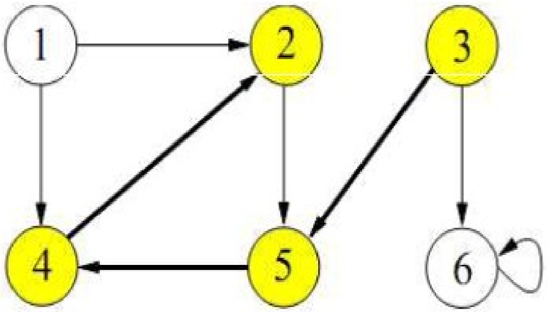
$$u=v_1, v=v_k, (v_{i-1}, v_1) \in E, \forall i=2,.., k$$

Longitud de un camino: número de arcos del camino.

Ej:

Camino desde 3 a 2 → <3,5,4,2>.

Longitud del camino → <3,5,4,2> es 3.



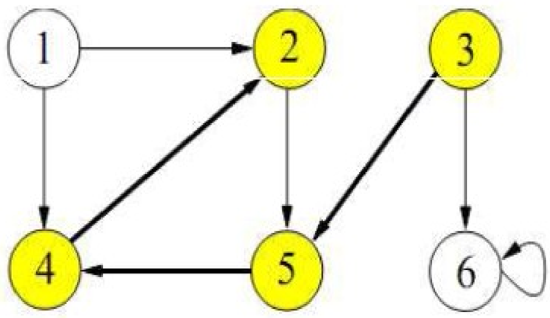
Descripción y terminología

Camino simple: camino en el que todos sus vértices, excepto, tal vez, el primero y el último, son distintos.

Ej:

(a) camino desde 3 a 2 → <3,5,4,2>

(b) camino desde 3 a 4 → <3,5,4,2,5,4>

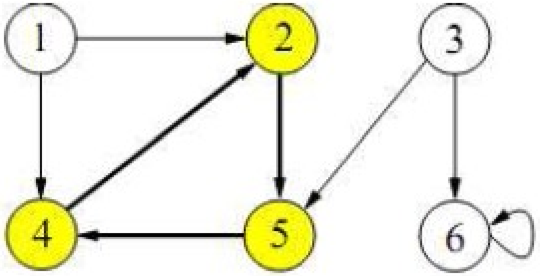


(a) es camino simple , (b) no lo es.

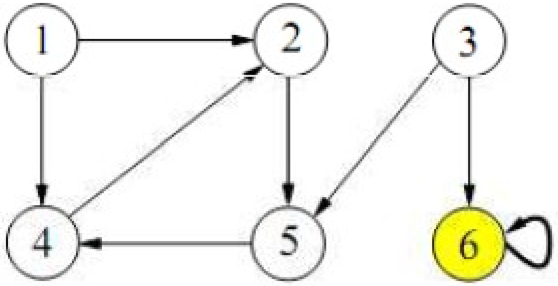
Descripción y terminología

Ciclo: camino simple desde v1, v2, ... , vk tal que v1=vk.

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



Bucle: ciclo de longitud 1.



Un bosque o grafo acíclico: grafo sin ciclos.

Descripción y terminología

Un grafo es **conexo** si entre cada dos nodos hay un camino.

Un grafo no dirigido es **conexo** si existe un camino desde cualquier vértice a cualquier otro.

Un grafo dirigido con esta propiedad se denomina **fuertemente conexo**.

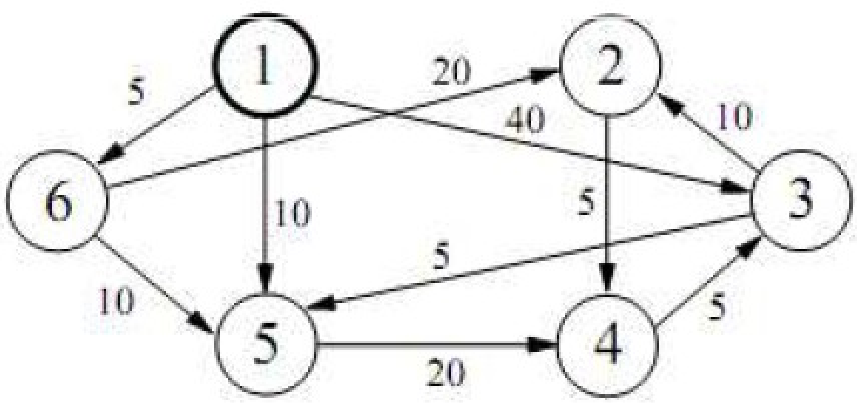
Si un grafo dirigido no es fuertemente conexo, pero el grafo subyacente (sin sentido en los arcos) es conexo, el grafo es **débilmente conexo**.

Descripción y terminología

v es **adyacente** a u si existe una arista (u,v) Є E.

v es **alcanzable** desde u, si existe un camino de u a v.

Un **grafo ponderado**, **pesado o con costos**: cada arco o arista tiene asociado un valor o etiqueta.



Representaciones

Representaciones

- Matriz de Adyacencias

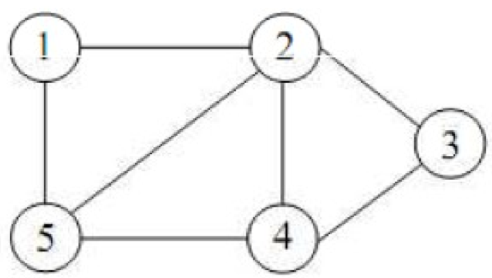
- Lista de Adyacencias

Representación: Matriz de Adyacencias

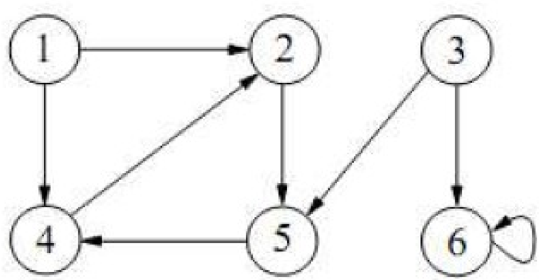
G =(V ,E): matriz A de dimensión |V |×|V |.

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



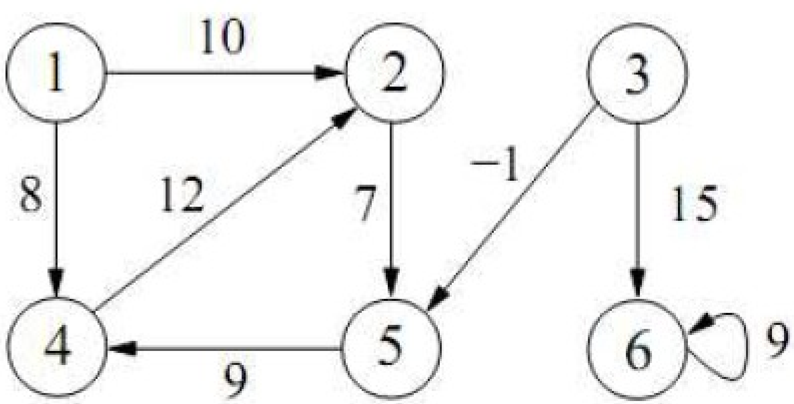
	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Representación: Matriz de Adyacencias

Representación aplicada a Grafos pesados

El peso de (i,j) se almacena en A (i, j).

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



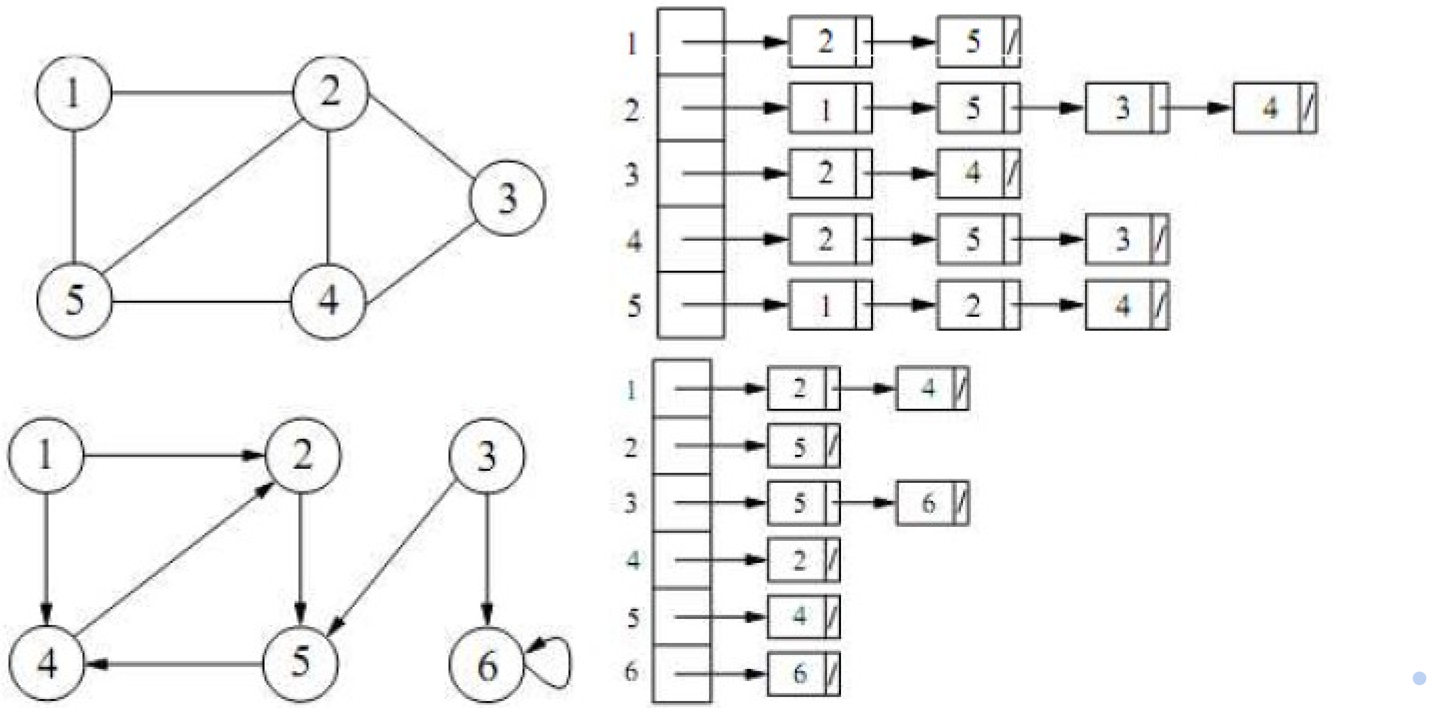
	1	2	3	4	5	6
1	0	10	0	8	0	0
2	0	0	0	0	7	0
3	0	0	0	0	−1	15
4	0	12	0	0	0	0
5	0	0	0	9	0	0
6	0	0	0	0	0	9

Representación: Lista de Adyacencias

G =(V ,E): vector de tamaño |V|.

Posición i → referencia a una lista de elementos (lista de adyacencia).

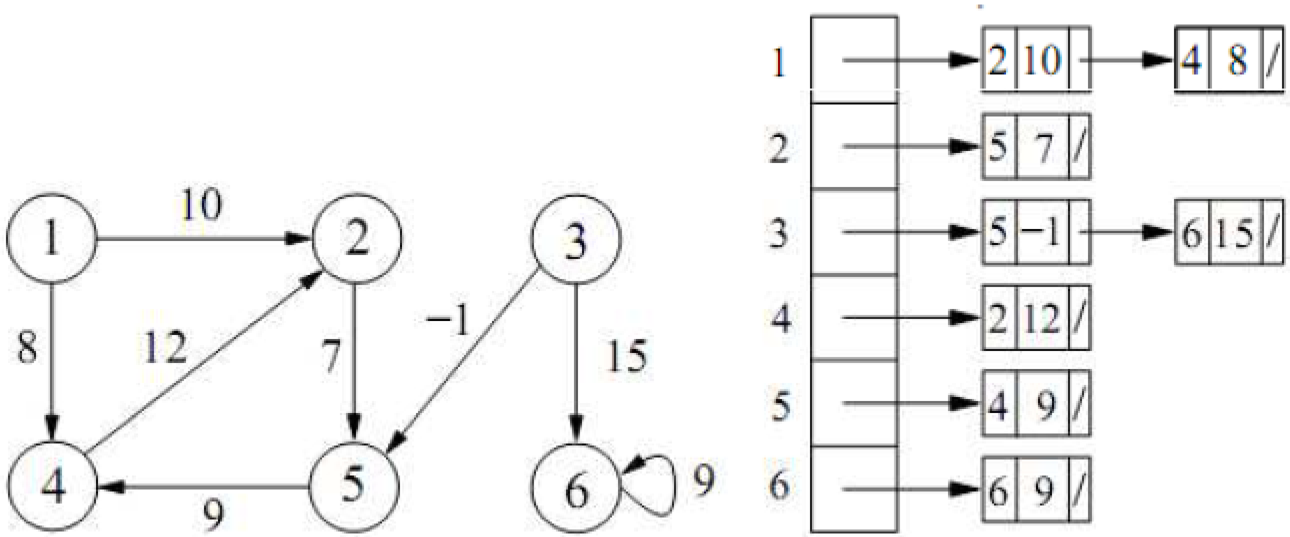
Los elementos de la lista son los vértices adyacentes a i



Representación: Lista de Adyacencias

Representación aplicada a Grafos pesados

El peso de (u,v) se almacena en el nodo de v de la lista de adyacencia de u.



Representaciones: Análisis

Matriz de Adyacencias:

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

Comprobar si una arista (u,v) pertenece a E → consultar posición A(u,v) con Costo de tiempo

$T(|V|,|E|) = O(1)$

Lista de Adyacencias:

Si G es dirigido, la suma de las longitudes de las listas de adyacencia será $|E|$ y si es no dirigido será $2|E|$.

Representación apropiada para grafos con $|E|$ menor que $|V|^2$.

Desv: Comprobar si una arista (u,v) pertenece a E → buscar v en la lista de adyacencia de u.

$T(|V|,|E|)$ será $O(|V|)$.

Recorridos

Recorridos

En profundidad : **DFS** (Depth First Search)

En amplitud : **BFS** (Breath First Search)

Ambos recorridos tienen orden **O(|V|+|E|)** .

Recorrido: DFS

Estrategia:

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. Para todo v adyacente a u,(u,v) Є E, si v no ha sido visitado, repetir recursivamente (3) y (4) para v.

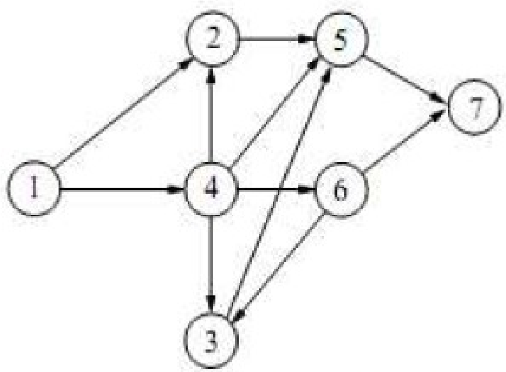
Recorrido: DFS

- Hasta que no se haya finalizado de explorar uno de los caminos no se comienza con el siguiente.

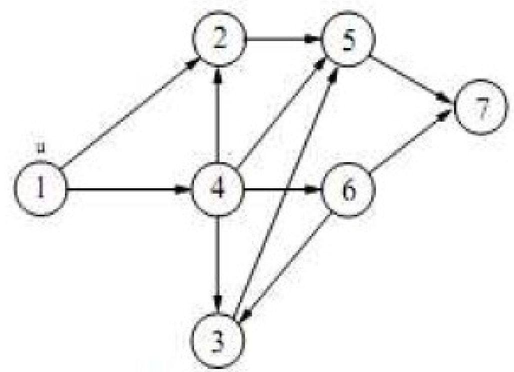
- Un camino deja de explorarse cuando se llega a un vértice ya visitado.

- Finalizar cuando se hayan visitado todos los nodos alcanzables desde u.

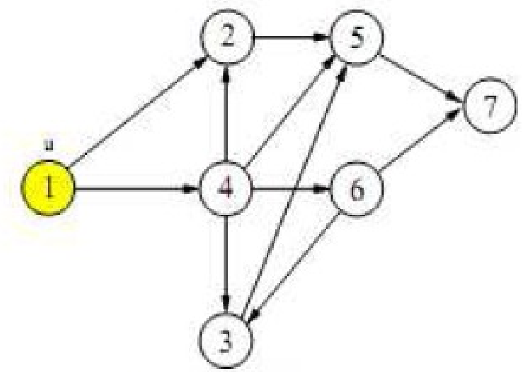
Ejemplo: DFS



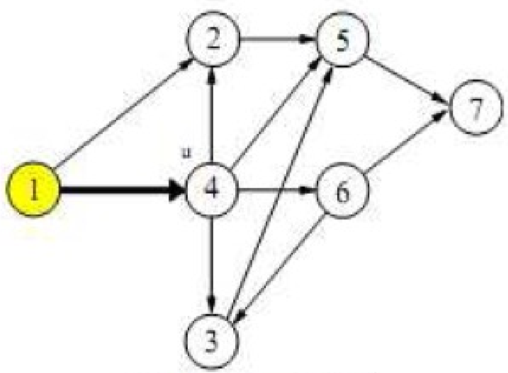
Rec_en_profund(G)



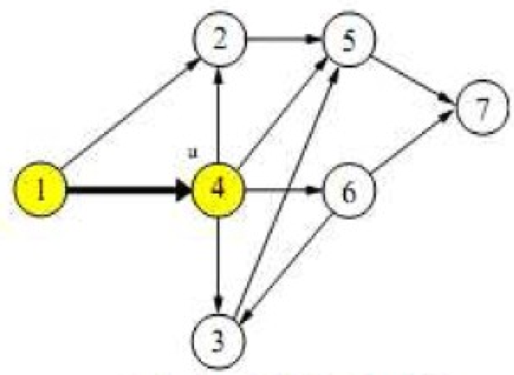
Visita_nodo(1)



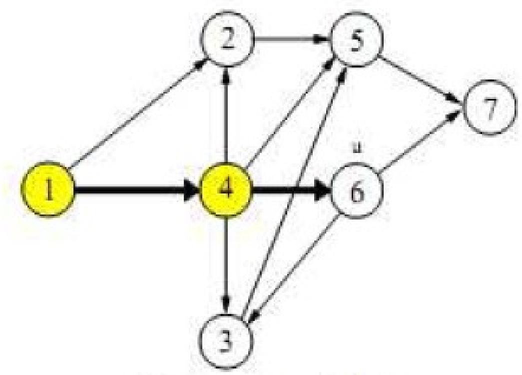
color[1]=AMARILLO



Visita_nodo(4)

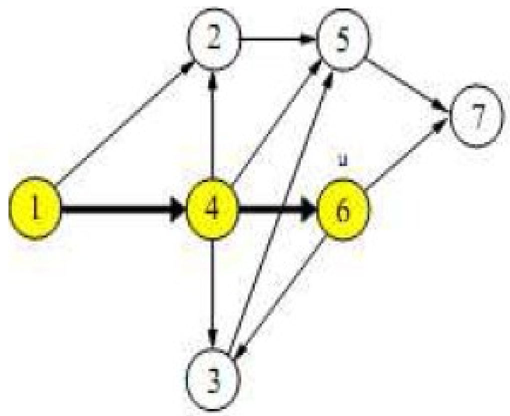


color[4]=AMARILLO

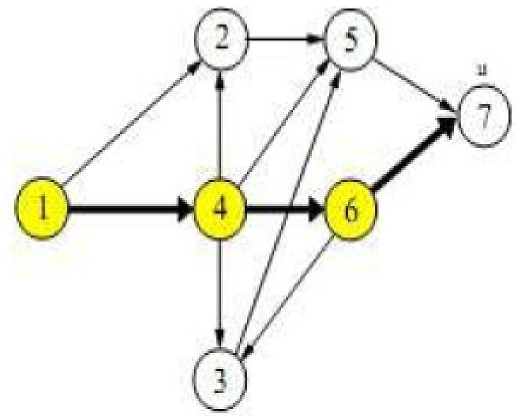


Visita_nodo(6)

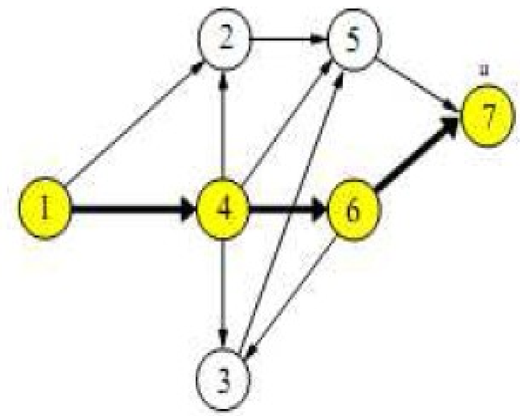
Ejemplo: DFS



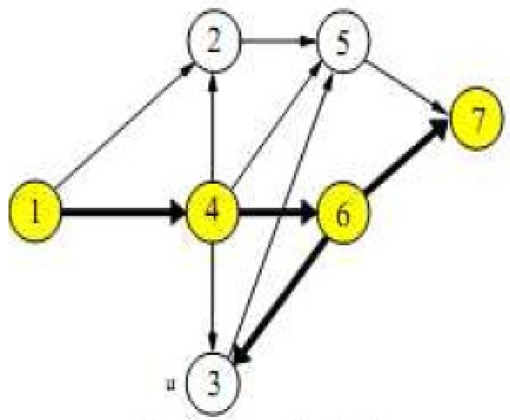
color[6]=AMARILLO



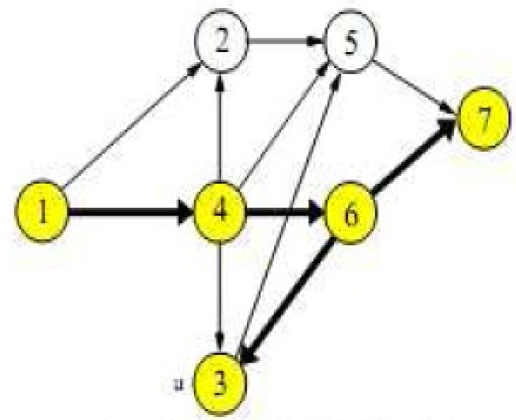
Visita_nodo(7)



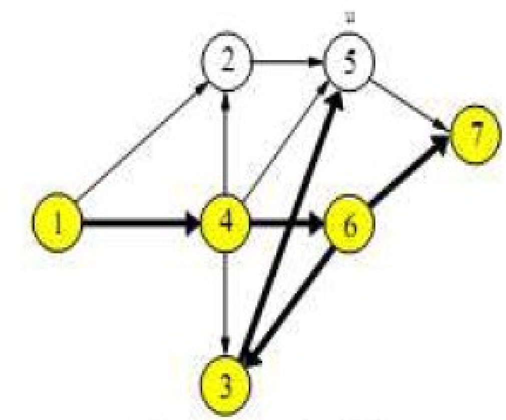
color[7]=AMARILLO



Visita_nodo(3)

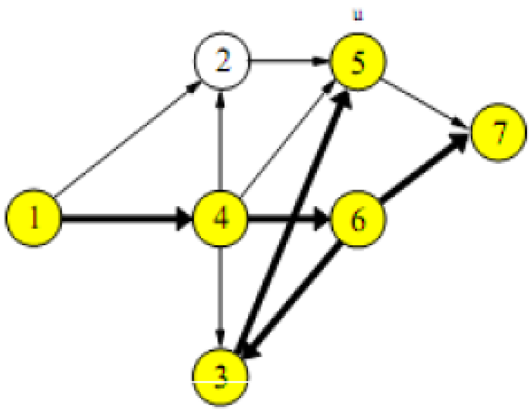


color[3]=AMARILLO

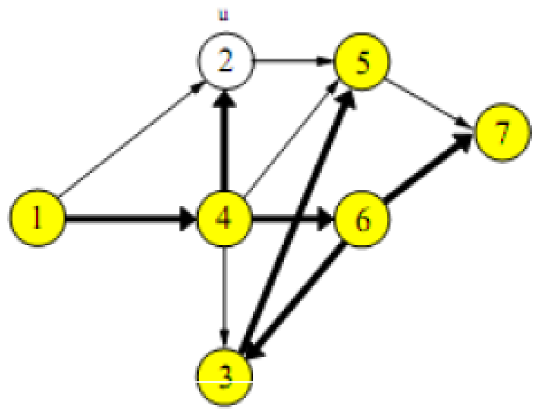


Visita_nodo(5)

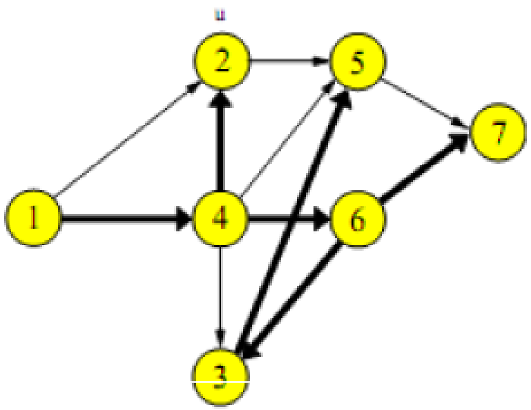
Ejemplo: DFS



color[5]=AMARILLO



Visita_nodo(2)



color[2]=AMARILLO

Recorrido: BFS

Estrategia:

Dado G = (V , E):

1. Marcar todos los vértices como no visitados.

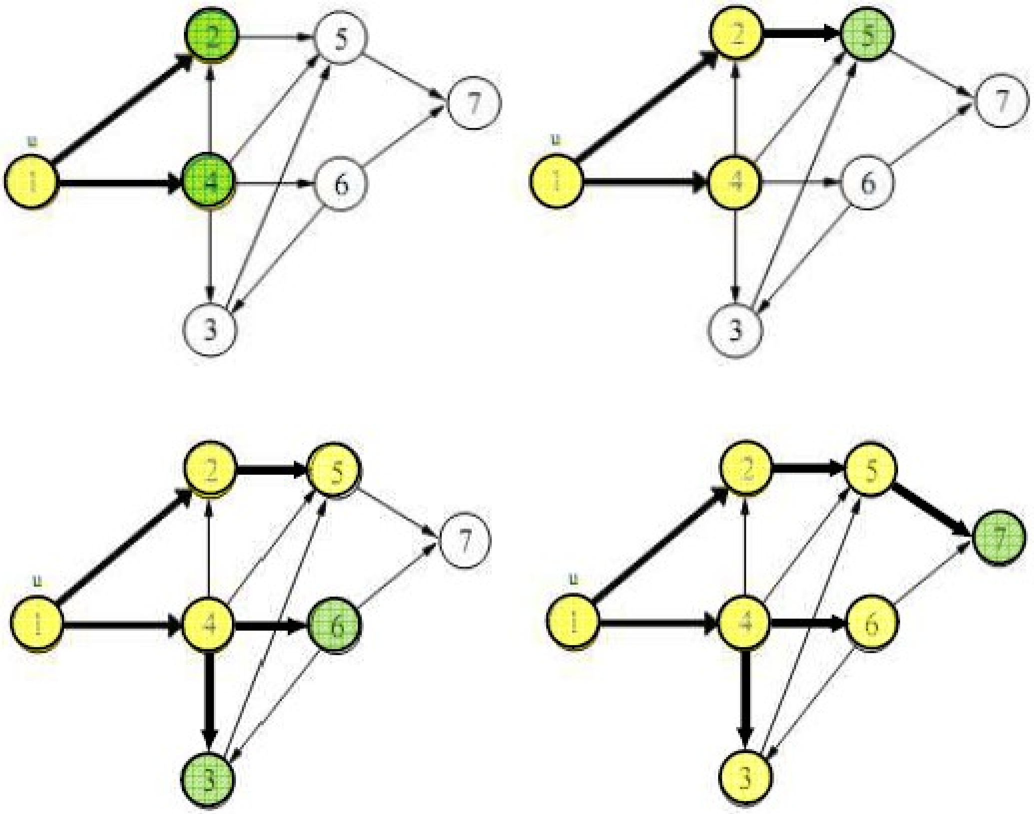
2. Elegir vértice u como punto de partida.

3. Marcar u como visitado.

4. Visitar cada uno de los vértices adyacentes a u.

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

Ejemplo: BFS



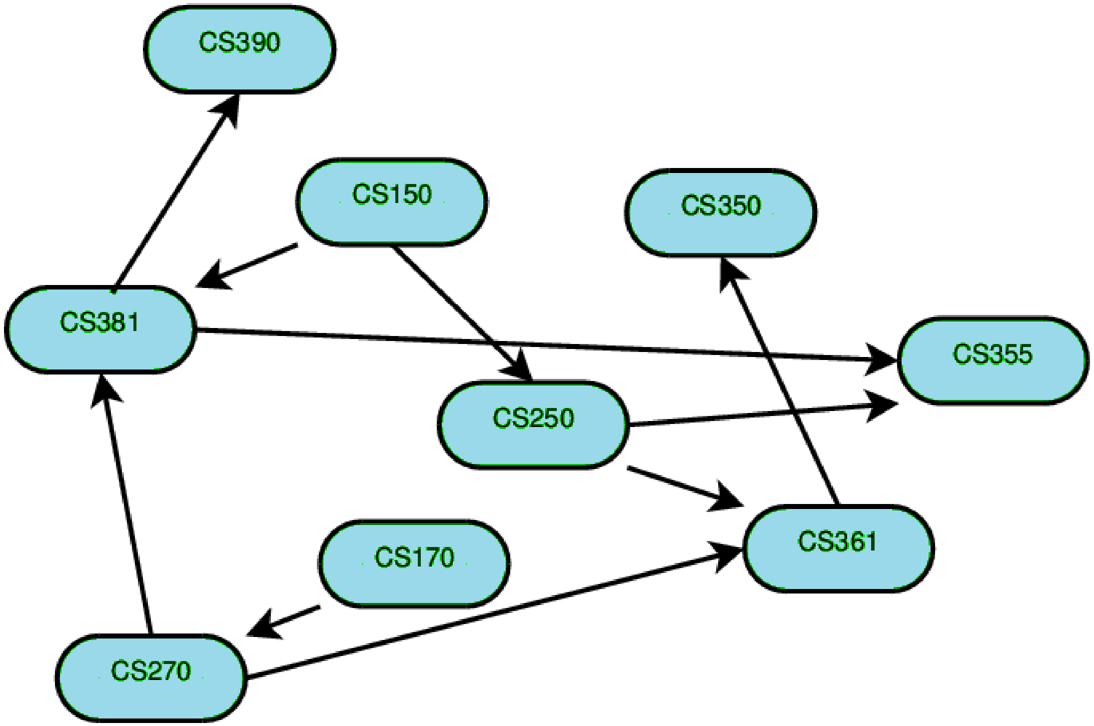
Ordenación topologica

Ordenación topológica - Definición

- La ordenación topológica es una permutación: $v_1, v_2, v_3, \ldots, v_{|V|}$ de los vértices, tal que si $(v_i,v_j) \in E$, $v_i \neq v_j$, entonces v_i precede a v_j en la permutación.
- La ordenación no es posible si G es cíclico.
- La ordenación topológica no es única.
- Una ordenación topológica es como una ordenación de los vértices a lo largo de una línea horizontal, con los arcos de izquierda a derecha.

Ordenación topológica

Ejemplo:



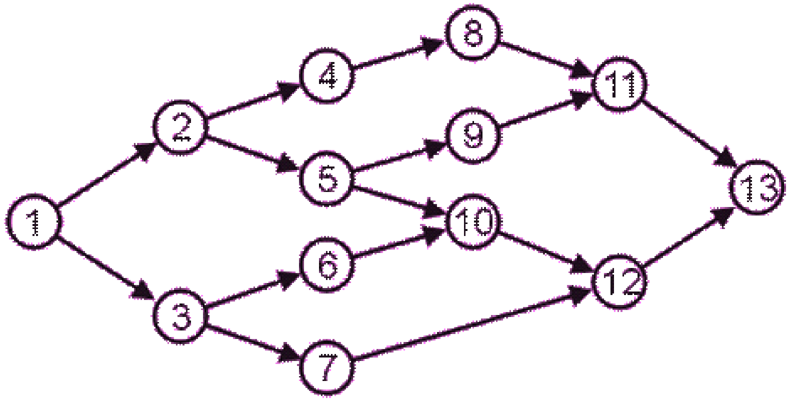
Cursos conectados por aristas que representan la *relación* de "prerrequisito"

Ordenación topológica

Dos ordenaciones válidas para el siguiente grafo:

1, 3, 2, 7, 6, 5, 4, 10, 9, 8, 12, 11, 13

1, 2, 4, 8, 5, 9, 11, 3, 6, 10, 7, 12, 13



Y hay muchas más……

Ordenación topológica – versión 1

Pasos generales:

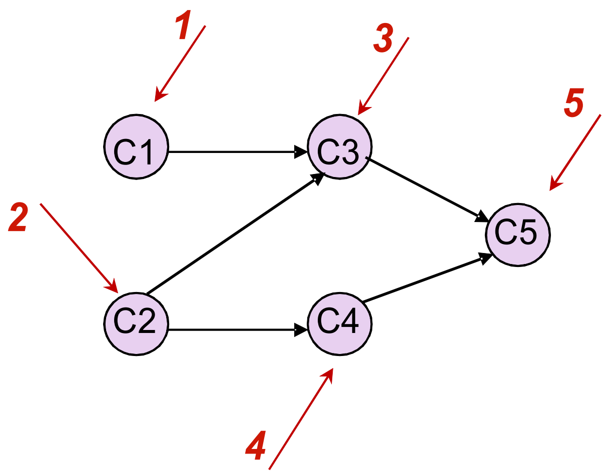
1. Seleccionar un vértice *v* con grado de entrada cero
2. Visitar *v*
3. "Eliminar" *v*, junto con sus aristas salientes
4. Repetir el paso 1 hasta seleccionar todos los vértices

Ordenación topológica – versión 1

→ *Tomando vértice con grado_in = 0 del vector Grado_in*

Grado_in

C1	C2	C3	C4	C5
0	0	2	1	2
0	**0**	**1**	1	2
0	0	**0**	**0**	2
0	0	0	**0**	**1**
0	0	0	0	**0**



Sort Topológico :

C1 C2 C3 C4 C5

Ordenación topológica – versión 2

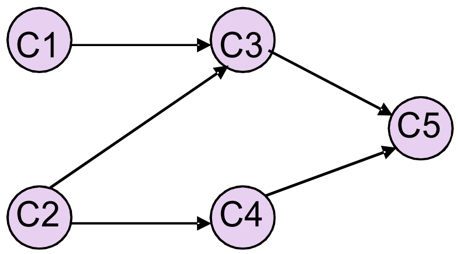
- *En esta versión el algoritmo utiliza un arreglo Grado_in en el que se almacenan los grados de entradas de los vértices y una pila P (o una cola Q) en donde se almacenan los vértices con grados de entrada igual a cero.*

Ordenación topológica – versión 2

→ *Tomando los vértices con grado_in = 0 de una Pila (o Cola)*

Grado_in

C1	C2	C3	C4	C5
0	**0**	2	1	2
0	0	**1**	**0**	2
0	0	1	0	**1**
0	0	**0**	0	**1**
0	0	0	0	**0**



Pila ***P*** : **C1** – **C2**

: C1 // C1 – **C4**

: C1 // C1

: // **C3**

: // **C5**

Sort Topológico :

C2 C4 C1 C3 C5

Ordenación topológica – versión 3

→ *En esta versión se aplica el recorrido en profundidad.*

- *Se realiza un recorrido DFS, marcando cada vértice en post-orden, es decir, una vez visitados todos los vértices a partir de uno dado, el marcado de los vértices en post-orden puede implementarse según una de las sig. opciones:*

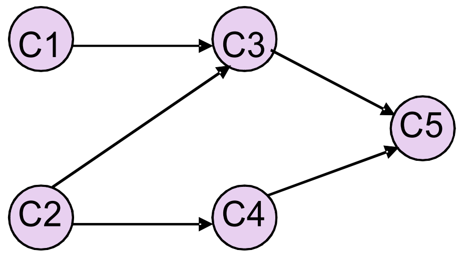
  a) numerándolos antes de retroceder en el recorrido; luego se listan los vértices según sus números de post-orden de mayor a menor.

  b) colocándolos en una pila P, luego se listan empezando por el tope.

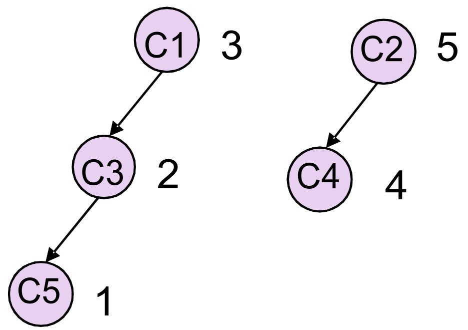
Ordenación topológica – versión 3

→ *Aplicando el recorrido en profundidad.*

*Opción **a)** - numerando los vértices*



Grafo dirigido acíclico



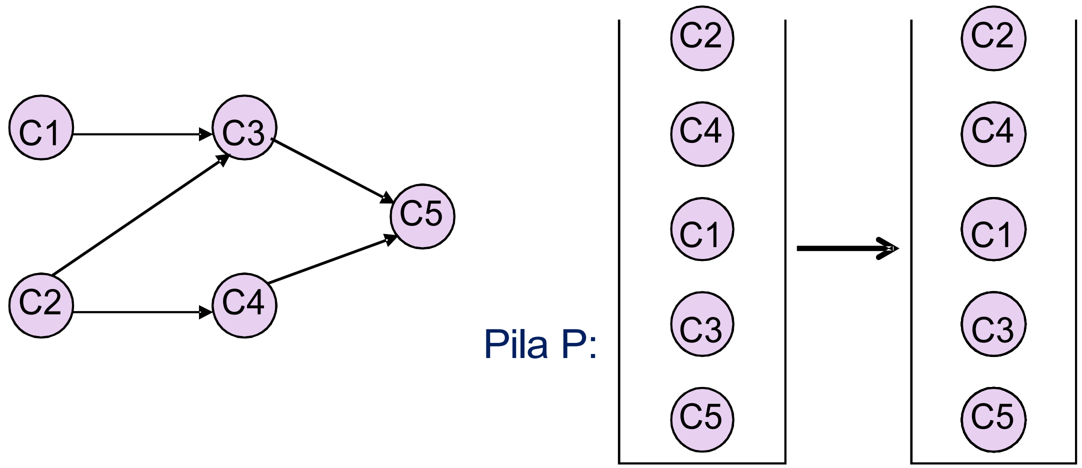
Aplico DFS a partir de un vértice cualquiera, por ejemplo C1

Ordenación Topológica: C2 C4 C1 C3 C5

Ordenación topológica – versión 3

→ *Aplicando el recorrido en profundidad.*

*Opción **b)** - apilando los vértices*



Ordenación Topológica: C2 C4 C1 C3 C5