UNIVERSIDAD NACIONAL
GUILLERMO BROWN

CLASE 13 - Unidad 9

Ordenamiento

ESTRUCTURAS DE DATOS (271)

Clase N. 13 Unidad 9.

Clase 13: AGENDA

AGENDA

- **Temario:**
  - Ordenamiento topológico.
  - Ejemplos de aplicación.
  - Distintas implementaciones.
  - Análisis de la eficiencia de cada uno.

- **Ejemplos en Lenguajes Python**
- **Temas relacionados y links de interés**
- **Práctica**
- **Cierre de la clase**

Clase 13: Ordenamiento topológico

Definición:

*El **orden topológico** de un digrafo G es un orden lineal de todos los vértices tales que si G contiene un arco (U,V), entonces U aparece antes de V en el ordenamiento.*

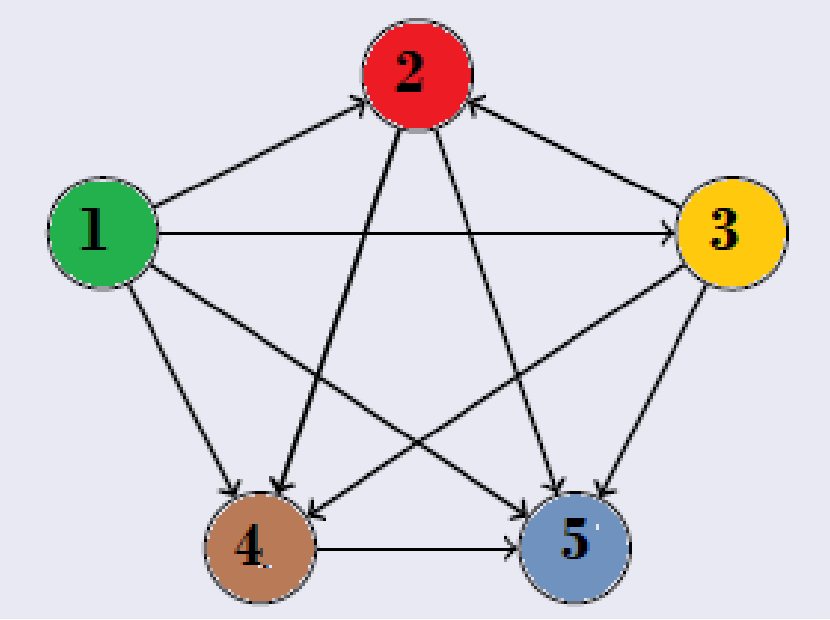
- Si el grafo tiene ciclos el orden topológico no es posible.
- Los digrafos son usados generalmente en aplicaciones para indicar la precedencia de eventos, el ordenamiento topológico devuelve la posible secuencia válida en la deben realizarse dichos eventos.
- Gráficamente se trata de poner todos los nodos en una línea de manera que sólo haya arcos hacia adelante.

El ordenamiento topológico es una adaptación simple pero útil de una búsqueda en profundidad.

Clase 13: Ordenamiento topológico

Ejemplo:

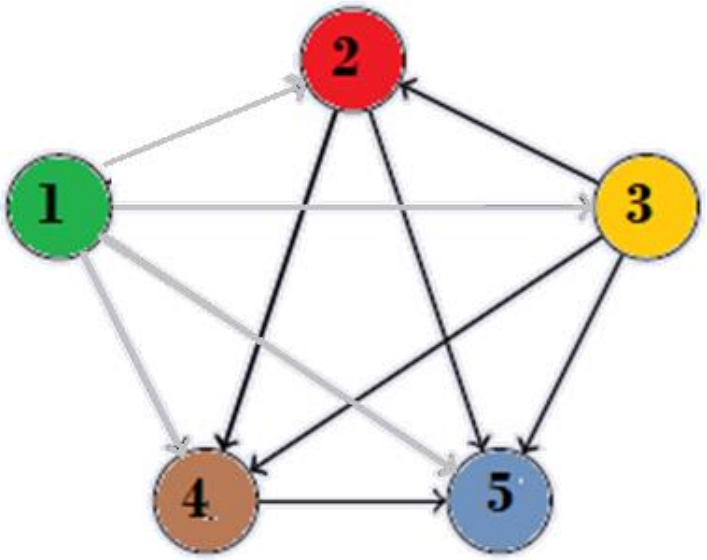
Una forma de encontrar un ordenamiento topológico es examinando todos los nodos del dígrafo y aquellos nodos que no tengan aristas incidentes sobre ellos son eliminados del grafo e introducidos en una cola (o lista), se repite este proceso hasta que no queden nodos en el grafo.



Sea G=<V,E> donde
V = {1,2,3,4,5}
E = {<1,1>,<2,2>,<3,3>,
<4,4>,<5,5>,<1,2>,
<1,3>,<3,2>,<2,4>,
<1,4>,<1,5>,<4,5>,
<3,5>,<2,5>,<3,4>}

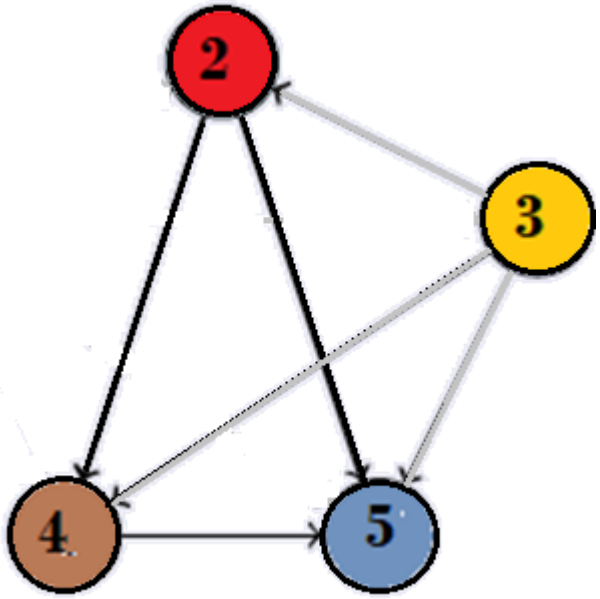
Clase 13: Ordenamiento topológico

Ejemplo:



1. Se escoge un vértice que no tenga lados incidentes y se coloca en una cola.

Escogemos el vértice **1**, lo eliminamos del grafo.(al eliminar este vértice, se eliminan todos las aristas que salen de él)

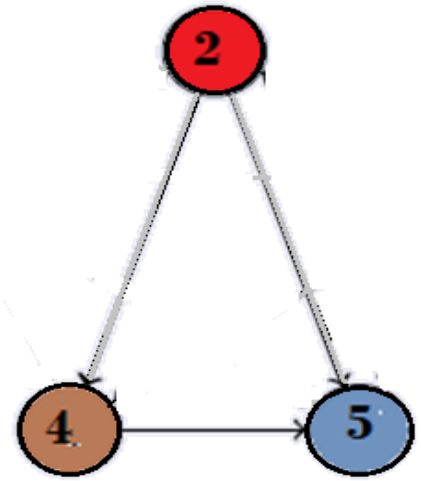


2. Del grafo resultante, se escoge un vértice que no tenga lados incidentes en él, se quita del grafo y se coloca en la cola.

Escogemos el vértice **3**, La cola que nos va quedando es **1 3**

Clase 13: Ordenamiento topológico

Ejemplo:



3. Del grafo resultante, se escoge un vértice que no tenga lados incidentes en él.

Se quita del grafo y se coloca en la cola **2,** La cola queda **1 3 2**

4. Ahora sólo quedan los vértices **4, 5** de los cuales se escoge el vértice **4** ya que no tiene lados incidentes en él y se coloca en la cola.
5. **Finalmente el orden topológico queda: 1 3 2 4 5**

Clase 13: Ordenamiento topológico

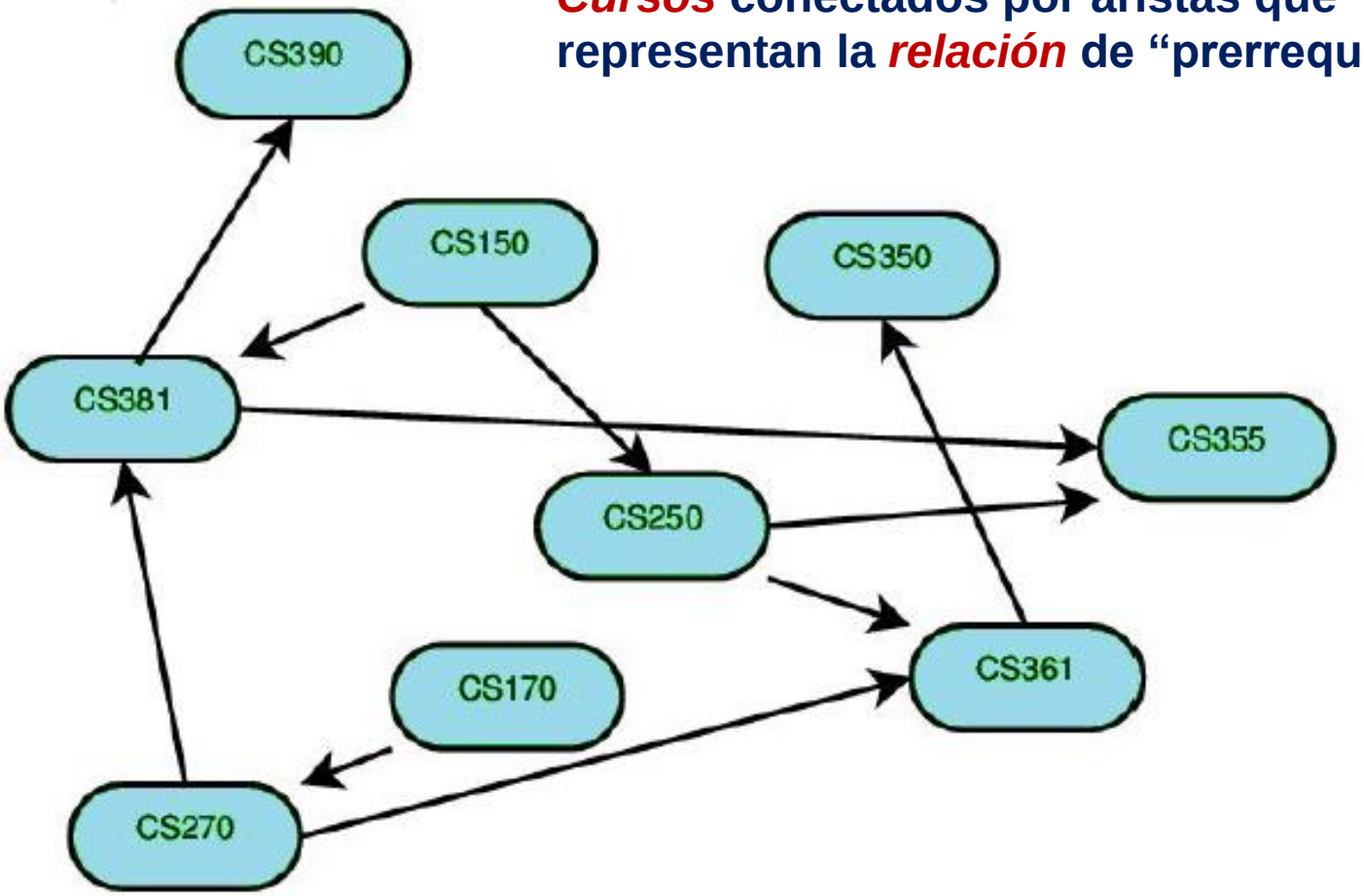
Aplicaciones:

- *Para indicar la precedencia entre eventos*
- *Para planificación de tareas*
- *Organización curricular*

Clase 13: Ordenamiento topológico

Aplicaciones:

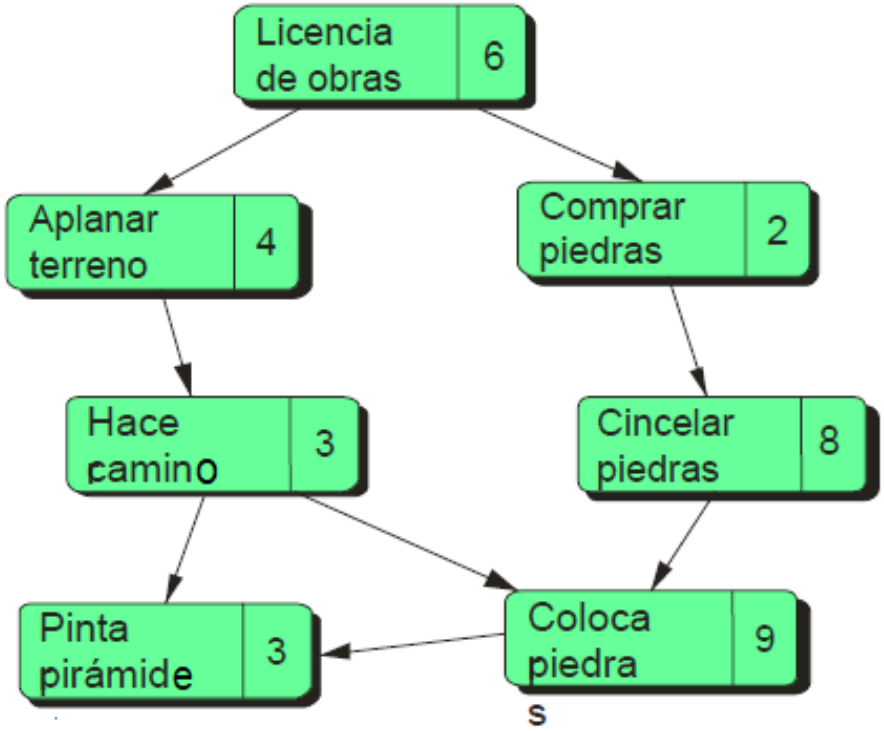
Cursos **conectados por aristas que representan la** ***relación*** **de "prerrequisito"**

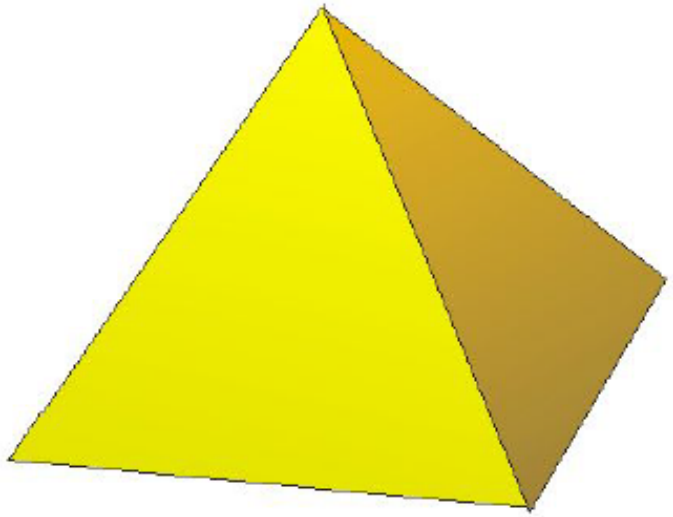


Clase 13: Ordenamiento topológico

Aplicaciones:

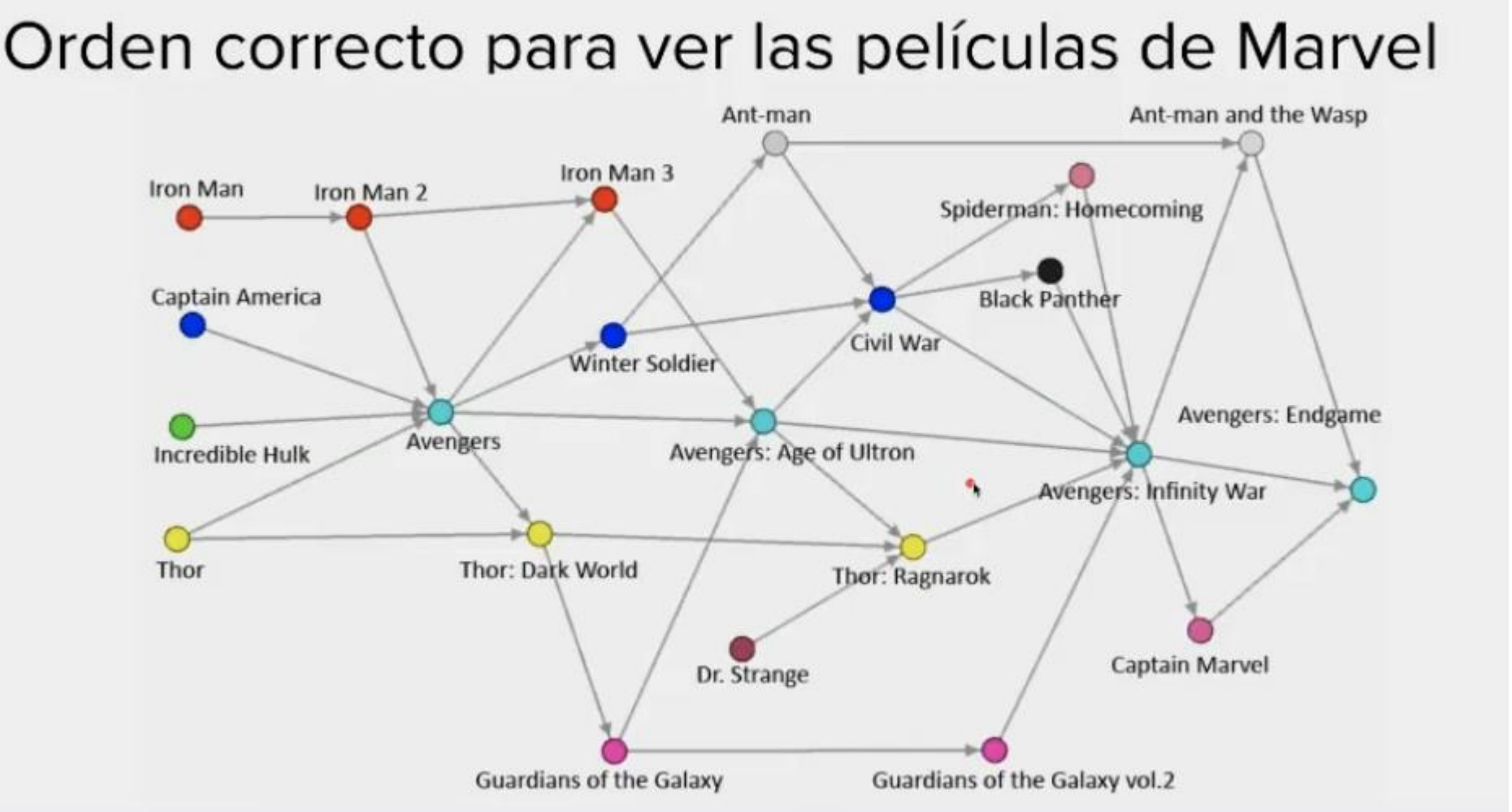
Planificación de Tareas





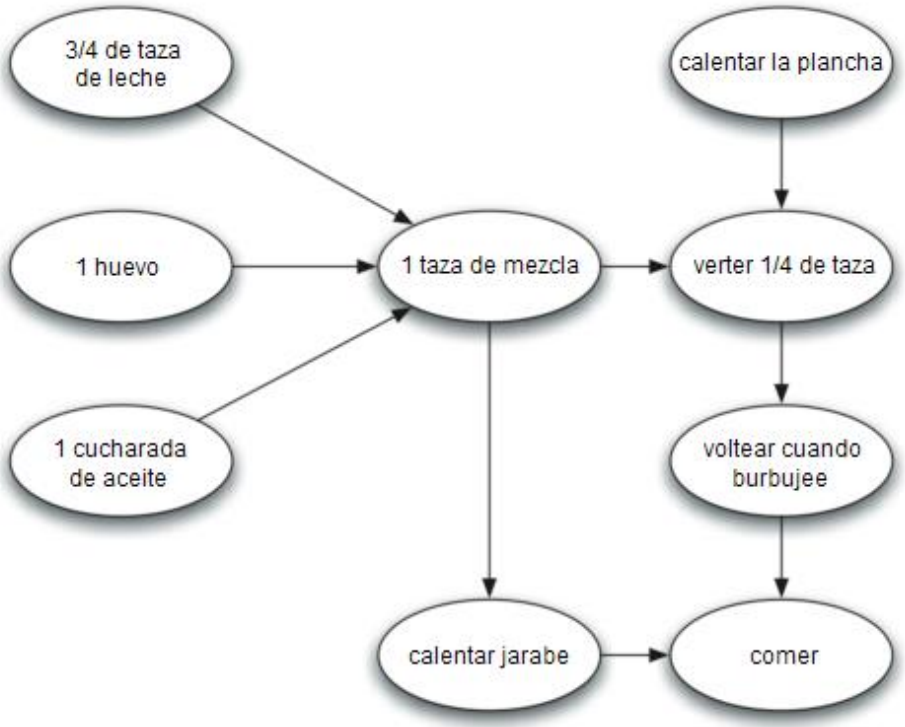
Clase 13: Ordenamiento topológico

Aplicaciones:



Clase 13: Ordenamiento topológico

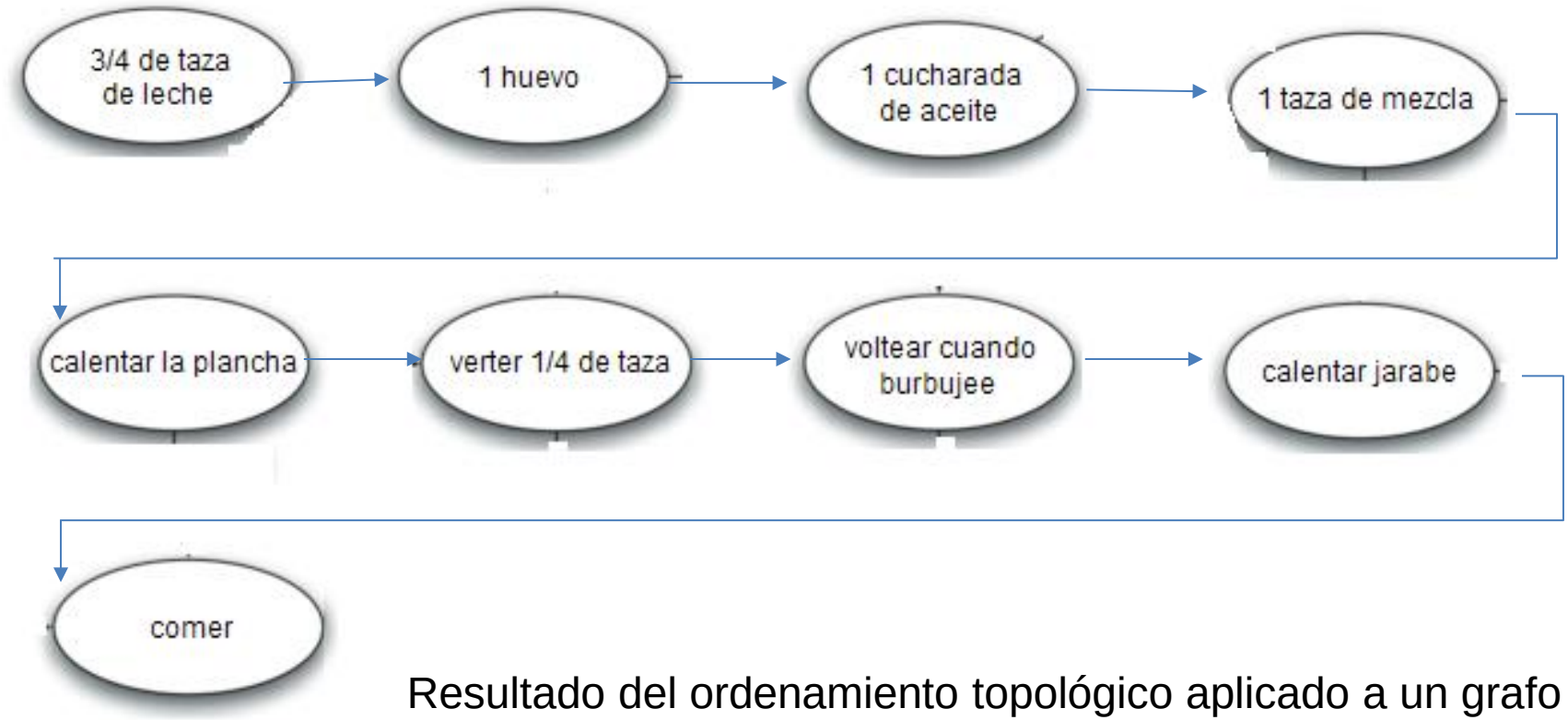
Ejemplos:



Dado el siguiente problema: batir un montón de panqueques. La receta es realmente muy simple: 1 huevo, 1 taza de mezcla de panqueques, 1 cucharada de aceite y 3/4 de una taza de leche. Para hacer panqueques usted debe calentar la plancha, mezclar todos los ingredientes y derramar la mezcla sobre una plancha caliente. Cuando los panqueques empiecen a burbujear, darlos vuelta y deje que se cocinen hasta que estén dorados en la parte de abajo. Antes de comer sus panqueques, usted querrá calentar un poco de jarabe dulce.

Clase 13: Ordenamiento topológico

Ejemplos:



Resultado del ordenamiento topológico aplicado a un grafo acíclico dirigido, siguiendo cada uno de los pasos descriptos anteriormente

Clase 13: Ordenamiento topológico

Algoritmo de Kahn pseudocódigo:

```
function TopologicalSort( Graph G ):
  for each node in G:
    calculate the indegree
  start = Node with 0 indegree
  G.remove(start)
  topological_list = [start]
  While node with 0 indegree present:
    topological_list.append(node)
    G.remove(node)
    Update Indegree of present nodes
  Return topological_list
```

O(|V|+|E|)

Clasificación topológica algoritmo DFS:

```
topologicalSort( )
initialize  stack and visited[n]
for each unvisited vertex v in graph:
   DFS(v):
      visited[v] = True
      for all unvisited neighbors w of vertex v:
            DFS(w)                    \\ recursively call DFS function
            push(v)                   \\ push vertex onto stack
until stack is empty:
      pop()                           \\ pop elements from stack
      list.append()                   \\ append to output list
```

O(|V|+|E|)

Clasificación topológica algoritmo DFS:

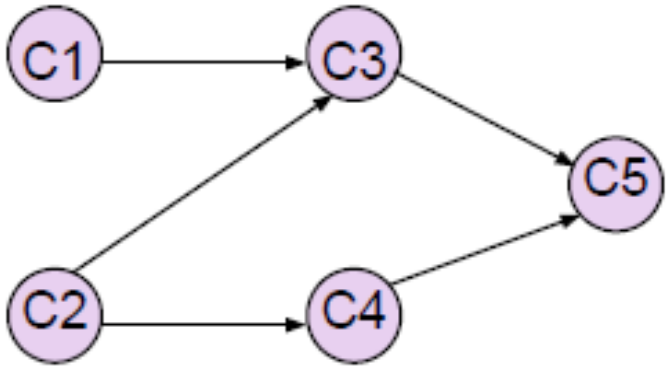
Se realiza un recorrido DFS, marcando cada vértice en post-orden, es decir, una vez visitados todos los vértices a partir de uno dado, el marcado de los vértices en post-orden puede implementarse según una de las sig. opciones:

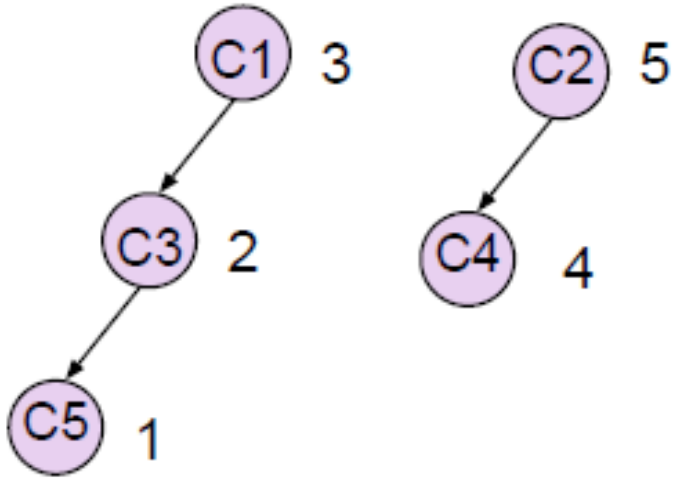
a) *numerándolos antes de retroceder en el recorrido; luego se listan los vértices según sus números de post-orden de mayor a menor.*

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

Clasificación topológica algoritmo DFS:

a) numerando los vértices





Grafo dirigido acíclico

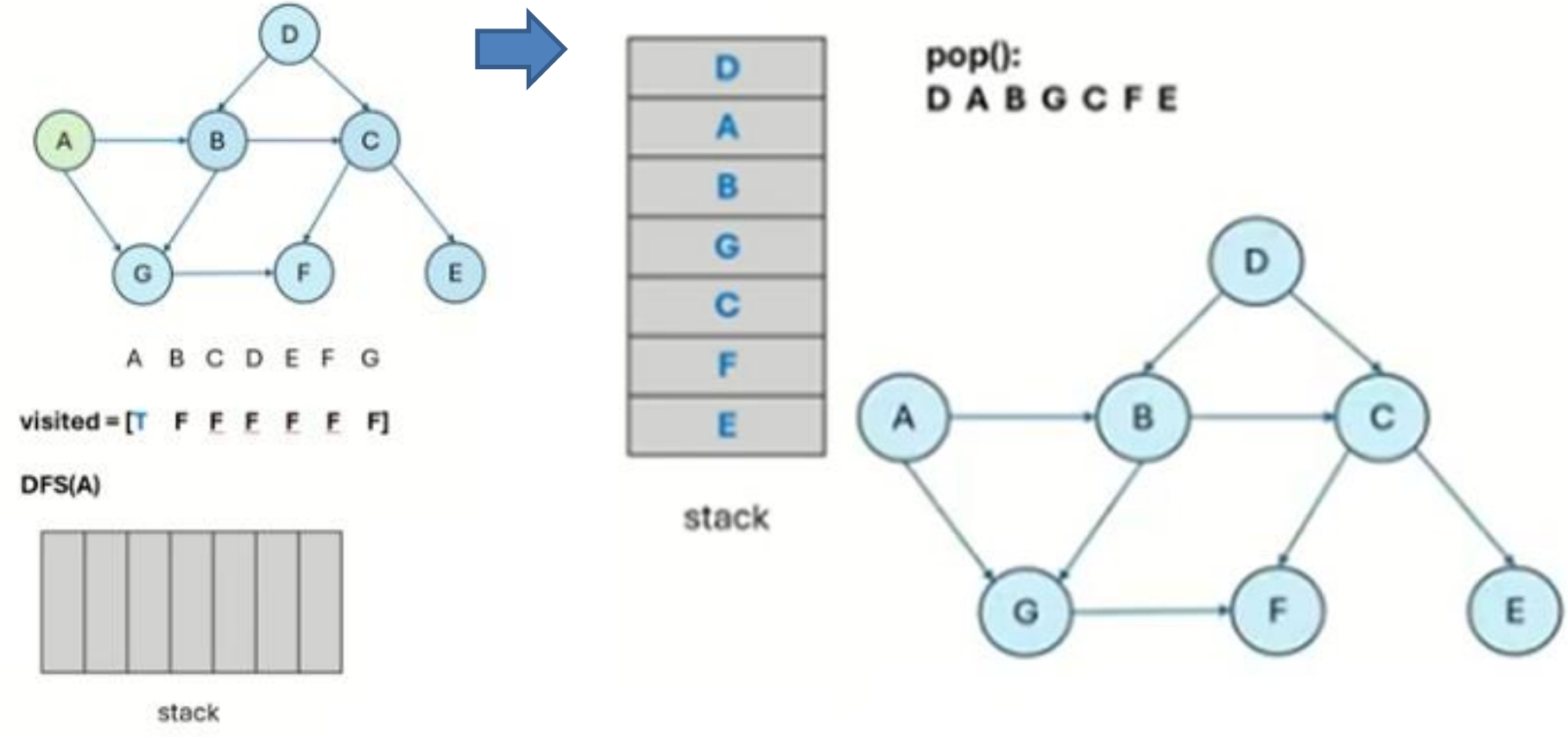
Aplico DFS a partir de un vértice cualquiera, por ejemplo C1

Ordenación Topológica: ***C2 C4 C1 C3 C5***

Clase 13: Ordenamiento topológico

Clasificación topológica algoritmo DFS:

b) *apilando los vértices*



Consultas

Temas a desarrollar la próxima clase

- Problema del camino mínimo: estudio de distintos casos.
- Su desarrollo para grafos pesados y no pesados; y grafos dirigidos y acíclicos. Algoritmos de Dijkstra y Floyd.
- Árbol generador mínimo. Algoritmos de Prim y Kruskal. Análisis del tiempo de ejecución de los algoritmos vistos.