

Diseño y análisis de algoritmos

ORDENACIÓN TOPOLÓGICA



Universidad
Rey Juan Carlos

Ordenamiento topológico

¿Por qué dos algoritmos?

- **El algoritmo visto en clase** es perfectamente válido para realizar la ordenación topológica
- **Sin embargo**, en el juez automático debemos ser deterministas (misma entrada \rightarrow misma salida)
- **Lo habitual** es pedir tareas que pueden realizarse en el mismo momento de tiempo en orden lexicográfico

Ordenamiento topológico

¿Por qué dos algoritmos?

- **El algoritmo visto en clase** depende del punto en el que se inicie el recorrido para mantener este orden
- **Está pensado para introducir** los algoritmos de **componentes fuertemente conexas** y **puntos de articulación**
- Por ello, os contamos esta alternativa

Ordenamiento topológico

Algoritmo alternativo

- La idea de esta opción es más simplista
- Se basa en contar las **aristas incidentes** a cada nodo para saber cuándo pueden ser realizadas las tareas que representan
- **Primero** se realizan las tareas con **0 aristas incidentes**
- Se actualiza el valor para cada nodo y se continúa

Ordenamiento topológico

Algoritmo alternativo - Pseudocódigo

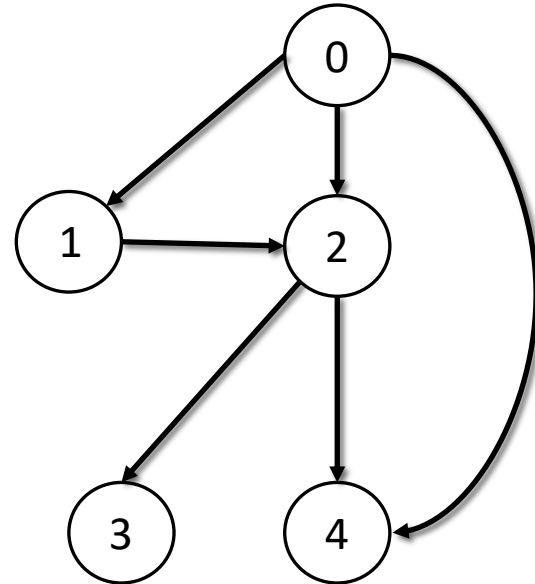
```
Topological_Sort(G):  
  for  $v \in G$   
    aristas_entrantes[v] = 0  
  for  $v \in G$   
    for  $a \in \text{vecinos}(v)$   
      aristas_entrantes[a] = aristas_entrantes[a] + 1  
  nodos_iniciales  $\leftarrow \emptyset$   
  for  $v \in G$   
    si aristas_entrantes[v] = 0 entonces:  
      nodos_iniciales  $\leftarrow \text{nodos\_iniciales} \cup v$   
  while nodos_iniciales  $\neq \emptyset$  :  
    origen = nodos_iniciales.pop(0)  
    recorrido = recorrido  $\cup$  origen  
    for  $a \in \text{vecinos}(\text{origen})$   
      aristas_entrantes[a] = aristas_entrantes[a] - 1  
      si aristas_entrantes[a] = 0 entonces:  
        nodos_iniciales  $\leftarrow \text{nodos\_iniciales} \cup a$ 
```

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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

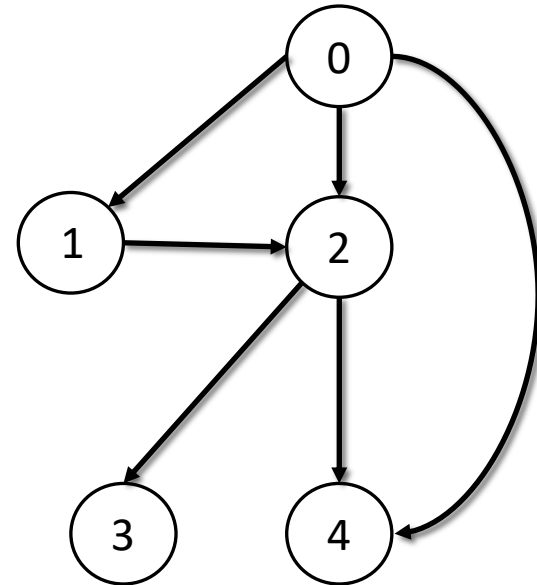


Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 1, 2, 1, 2]

nodos_iniciales = [0]

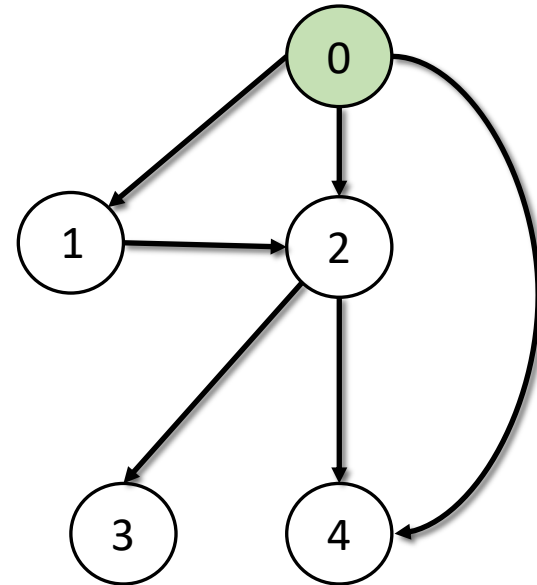
recorrido = []

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 0, 1, 1, 1]

nodos_iniciales = [1]

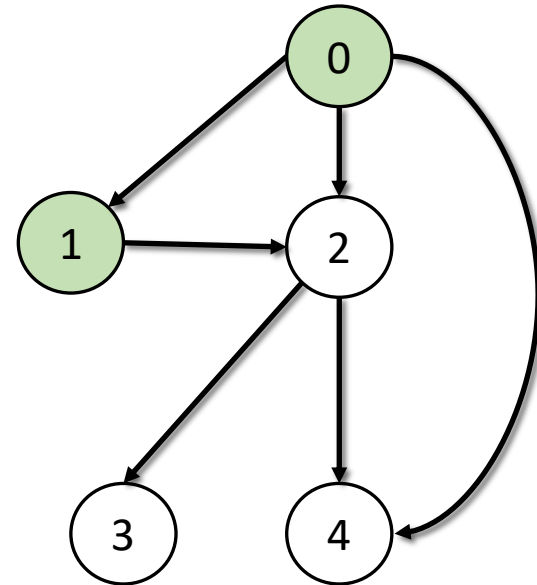
recorrido = [0]

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 0, 0, 1, 1]

nodos_iniciales = [2]

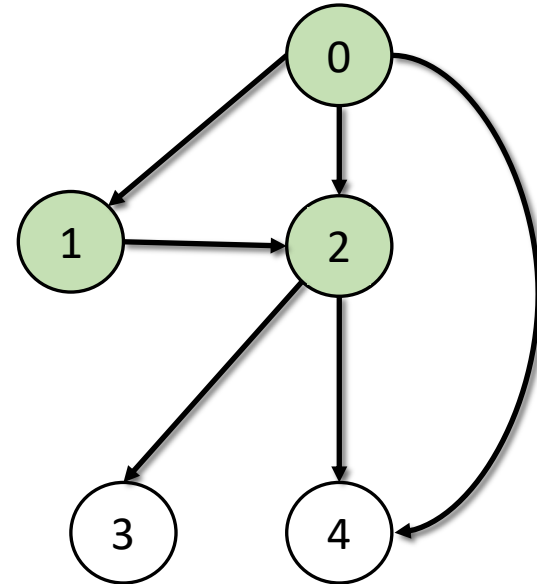
recorrido = [0, 1]

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 0, 0, 0, 0, 0]

nodos_iniciales = [3, 4]

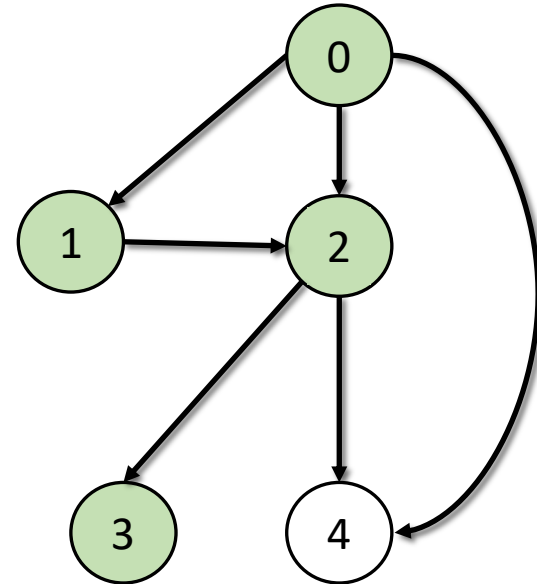
recorrido = [0, 1, 2]

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 0, 0, 0, 0]

nodos_iniciales = [4]

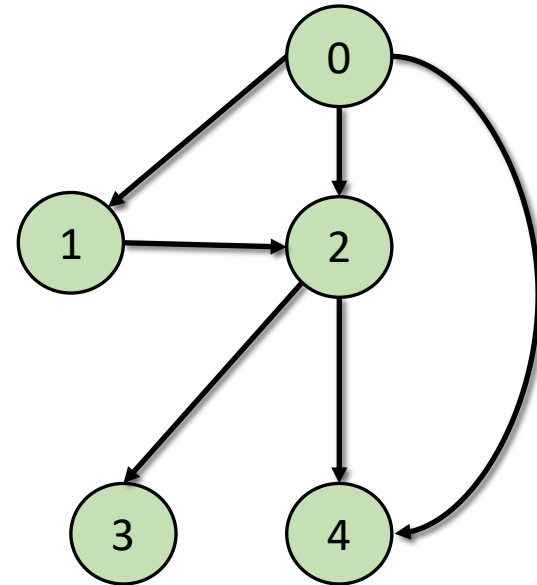
recorrido = [0, 1, 2, 3]

Ordenamiento topológico

Algoritmo alternativo

Ejemplo de entrada

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



aristas_entrantes = [0, 0, 0, 0, 0]

nodos_iniciales = []

recorrido = [0, 1, 2, 3, 4]

Ordenamiento topológico

Algoritmo alternativo

- En este caso, hemos obtenido el orden lexicográfico sin tener que hacer nada especial
- Os corresponde a vosotros encontrar la modificación del algoritmo para que siempre se cumpla

Diseño y análisis de algoritmos

ORDENACIÓN TOPOLÓGICA



Universidad
Rey Juan Carlos