

ORDENACIÓN TOPOLÓGICA

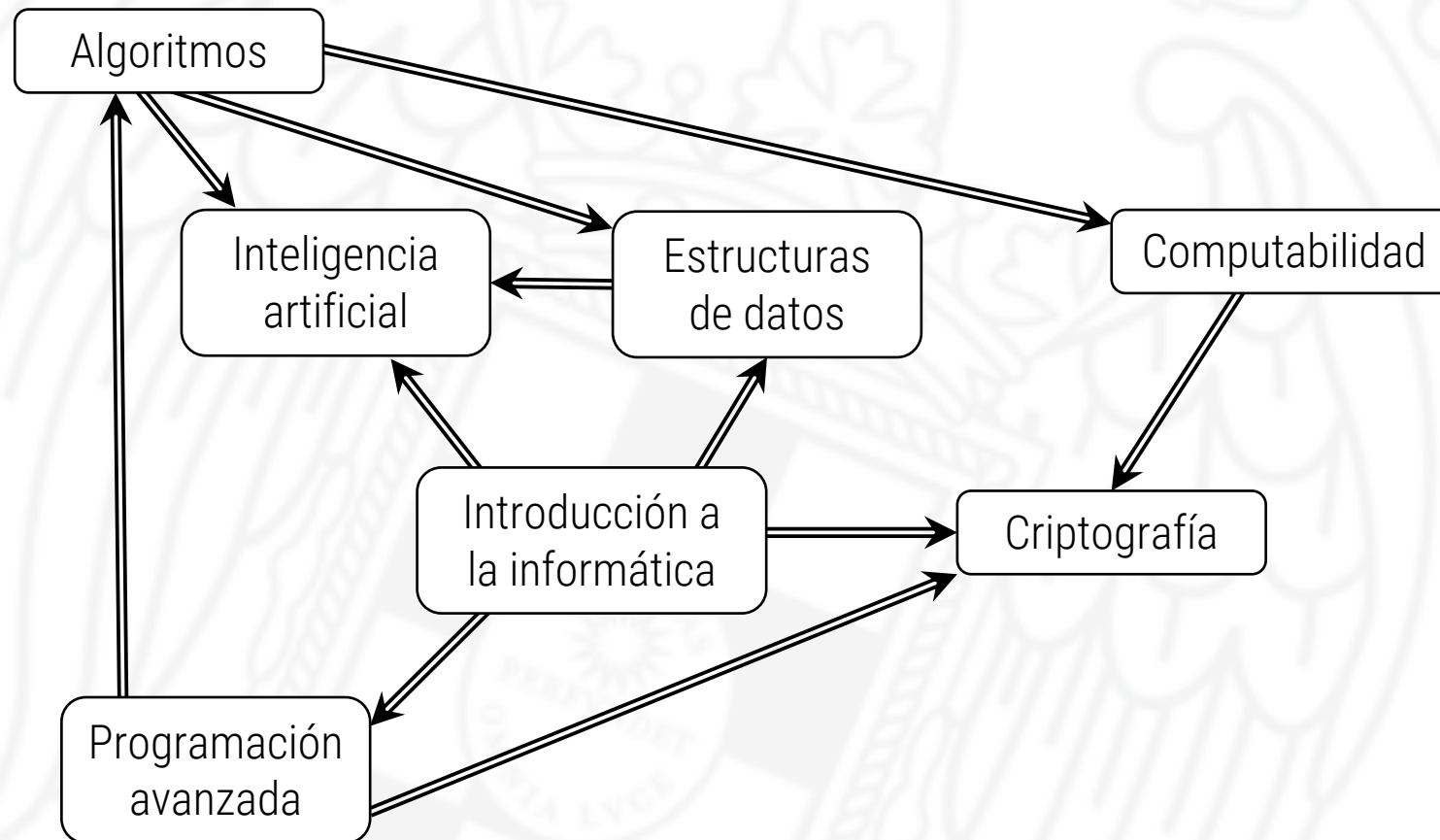


U N I V E R S I D A D
COMPLUTENSE
M A D R I D

ALBERTO VERDEJO

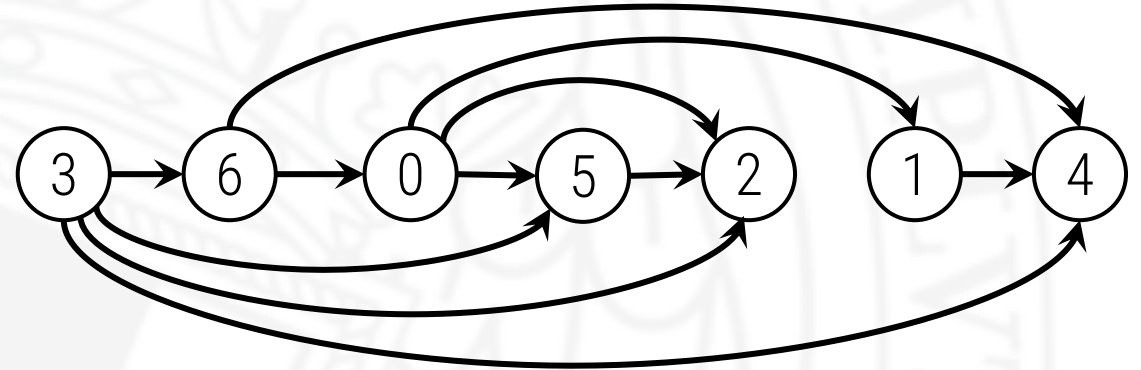
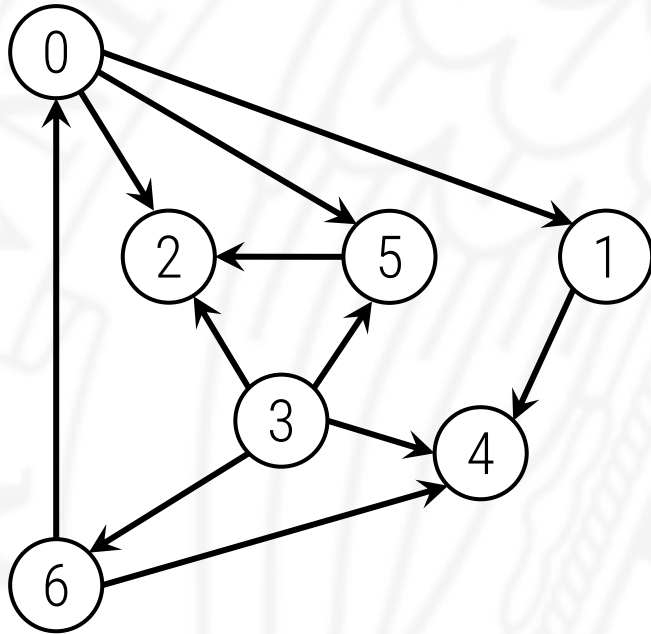
Planificación de tareas con precedencia

- Dado un conjunto de tareas a realizar con restricciones de precedencia, ¿en qué orden deberíamos planificarlas?



Ordenación topológica

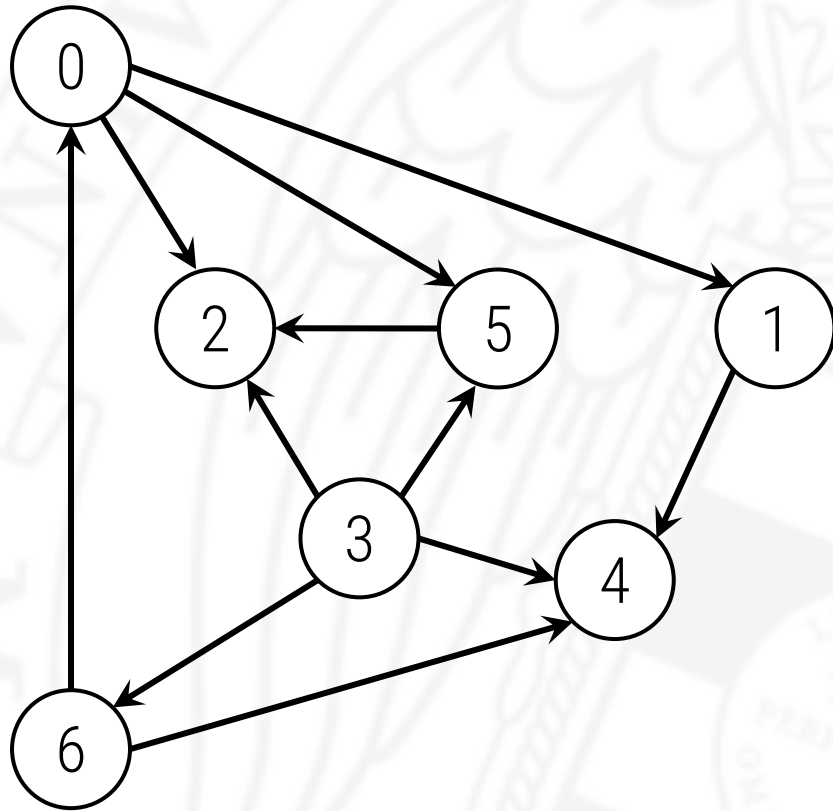
- Ordenar los vértices de forma que todas las aristas vayan de un vértice a otro posterior en la ordenación.



- Imprescindible que no haya ciclos (DAG).

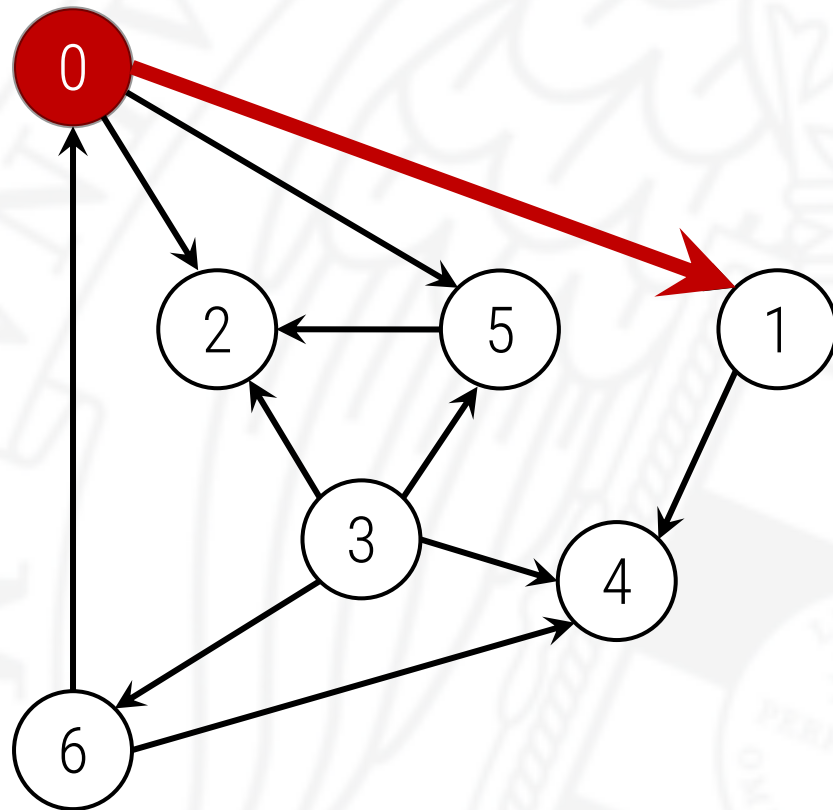
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

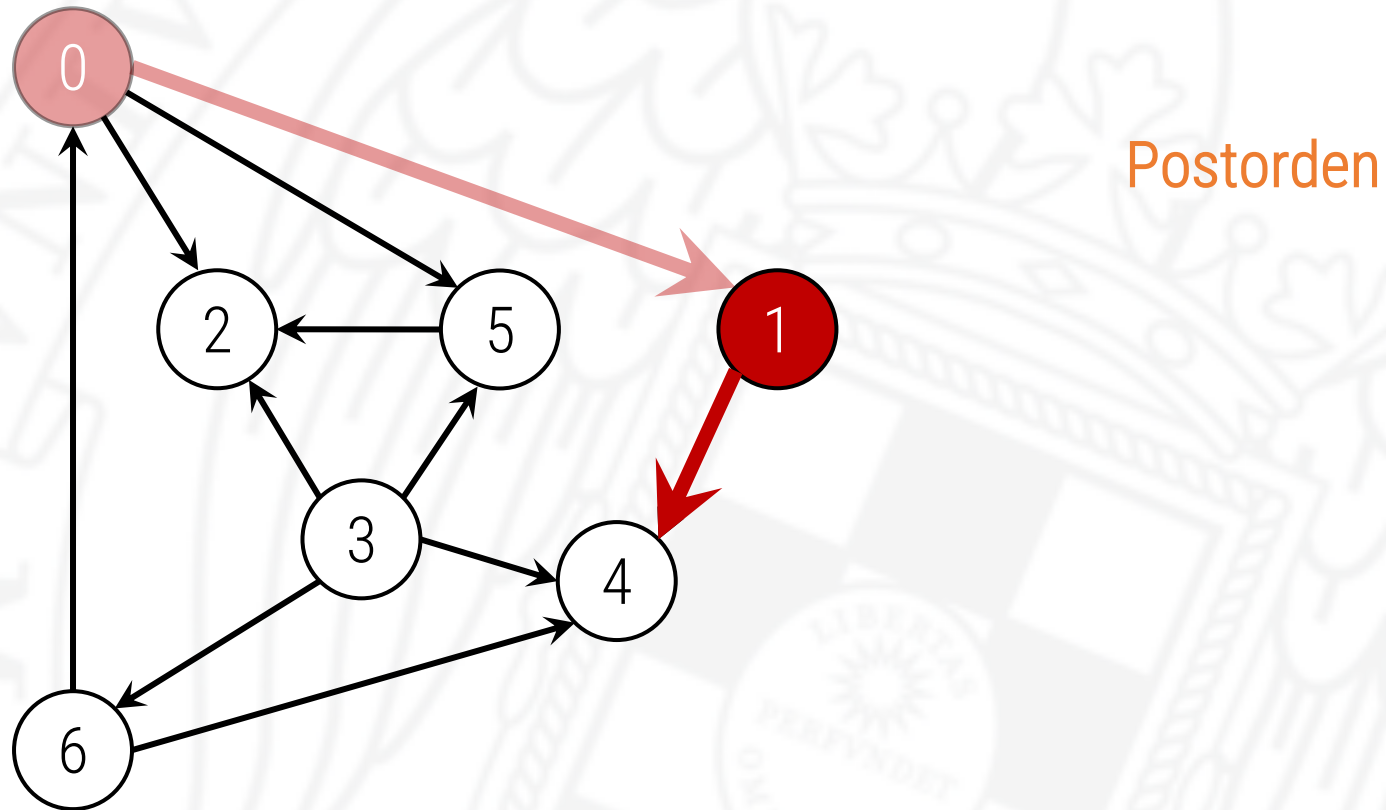
- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

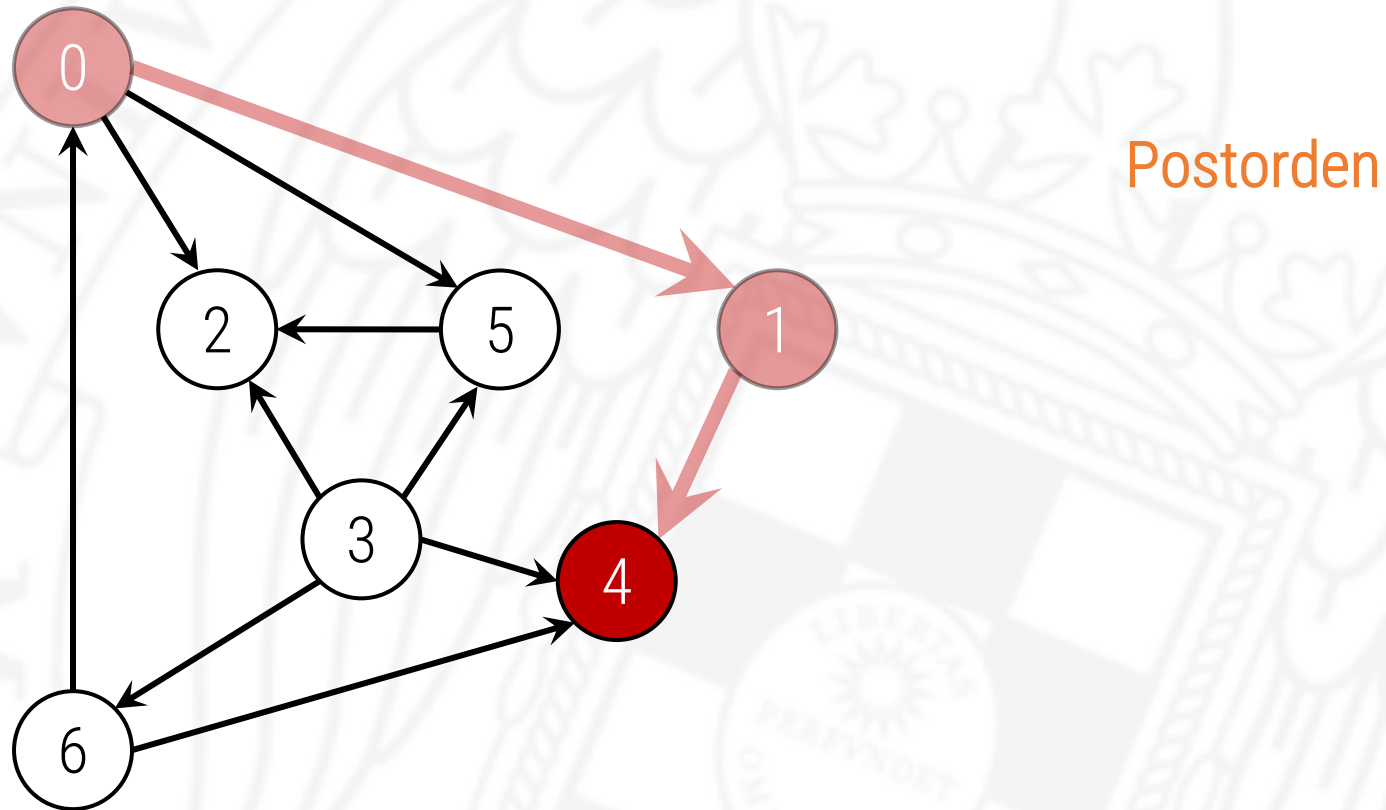
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



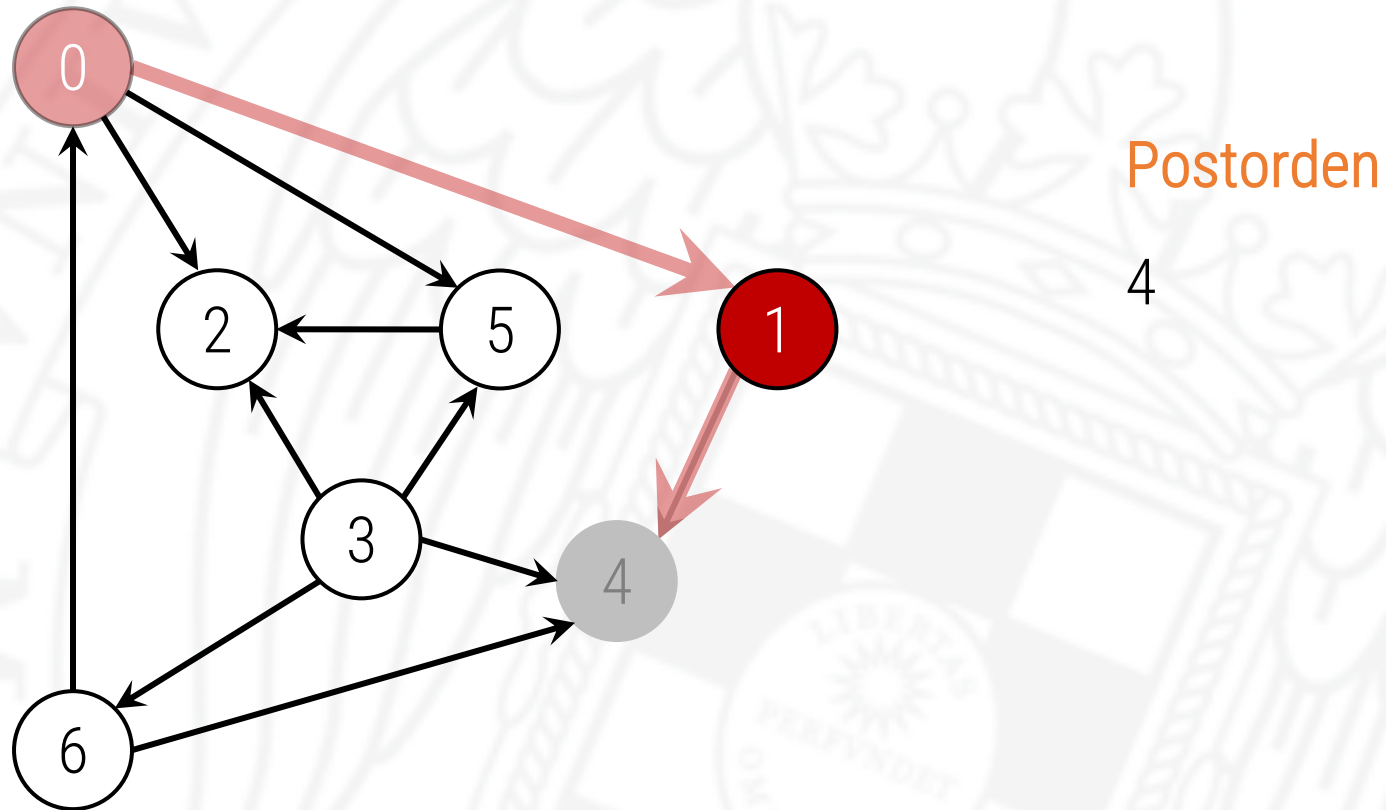
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



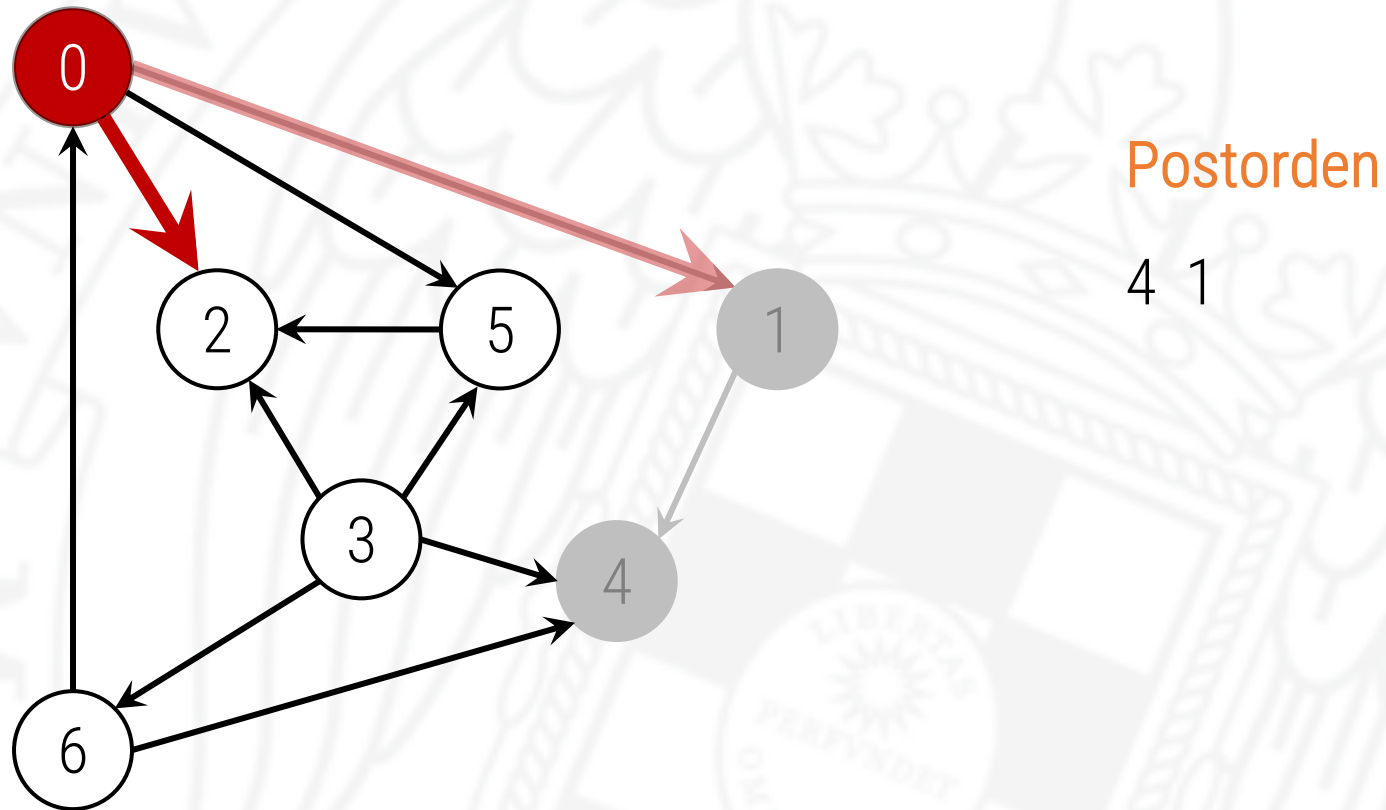
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



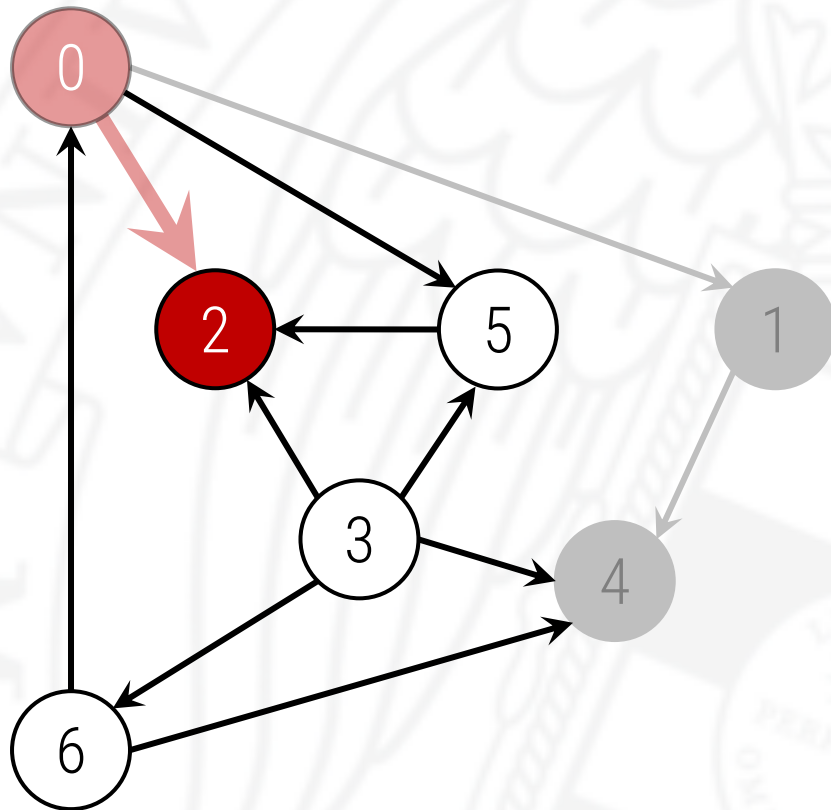
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

- Añadir el vértice al orden tras las llamadas recursivas.

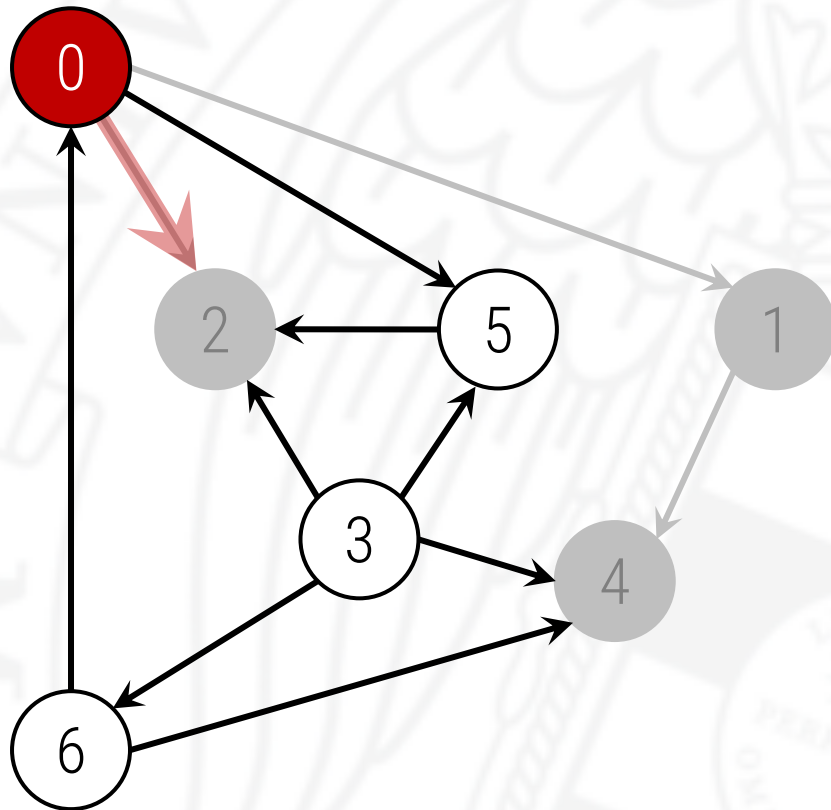


Postorden

4 1

Postorden

- Añadir el vértice al orden tras las llamadas recursivas.

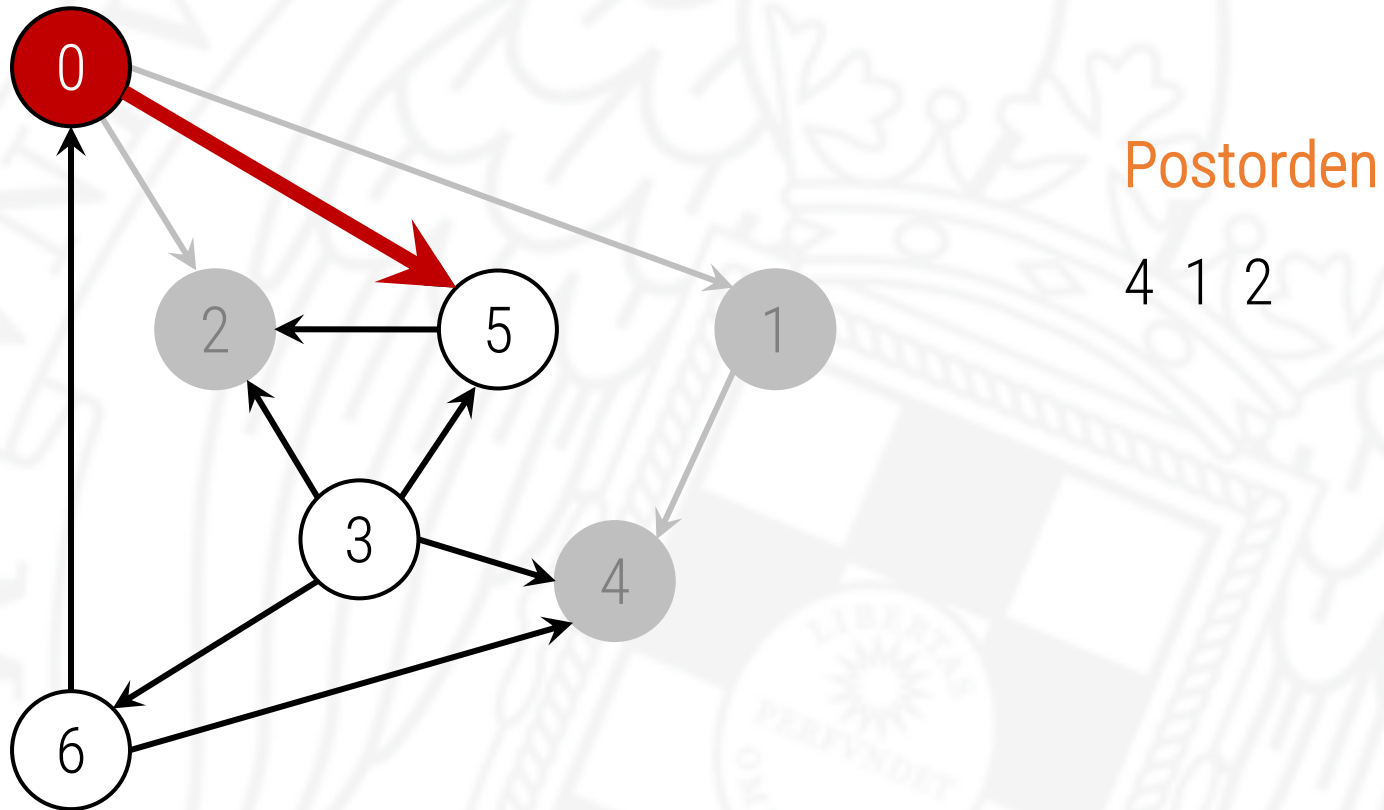


Postorden

4 1 2

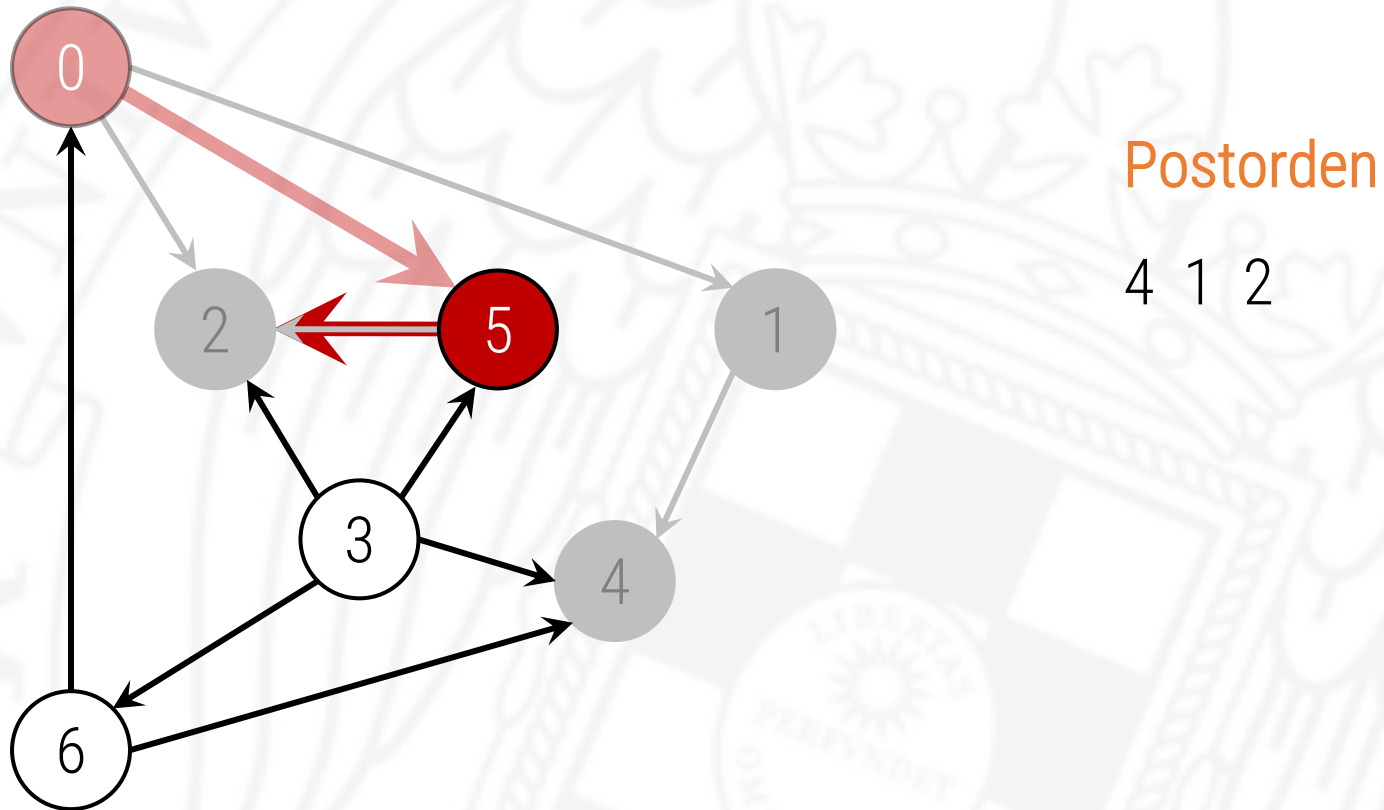
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



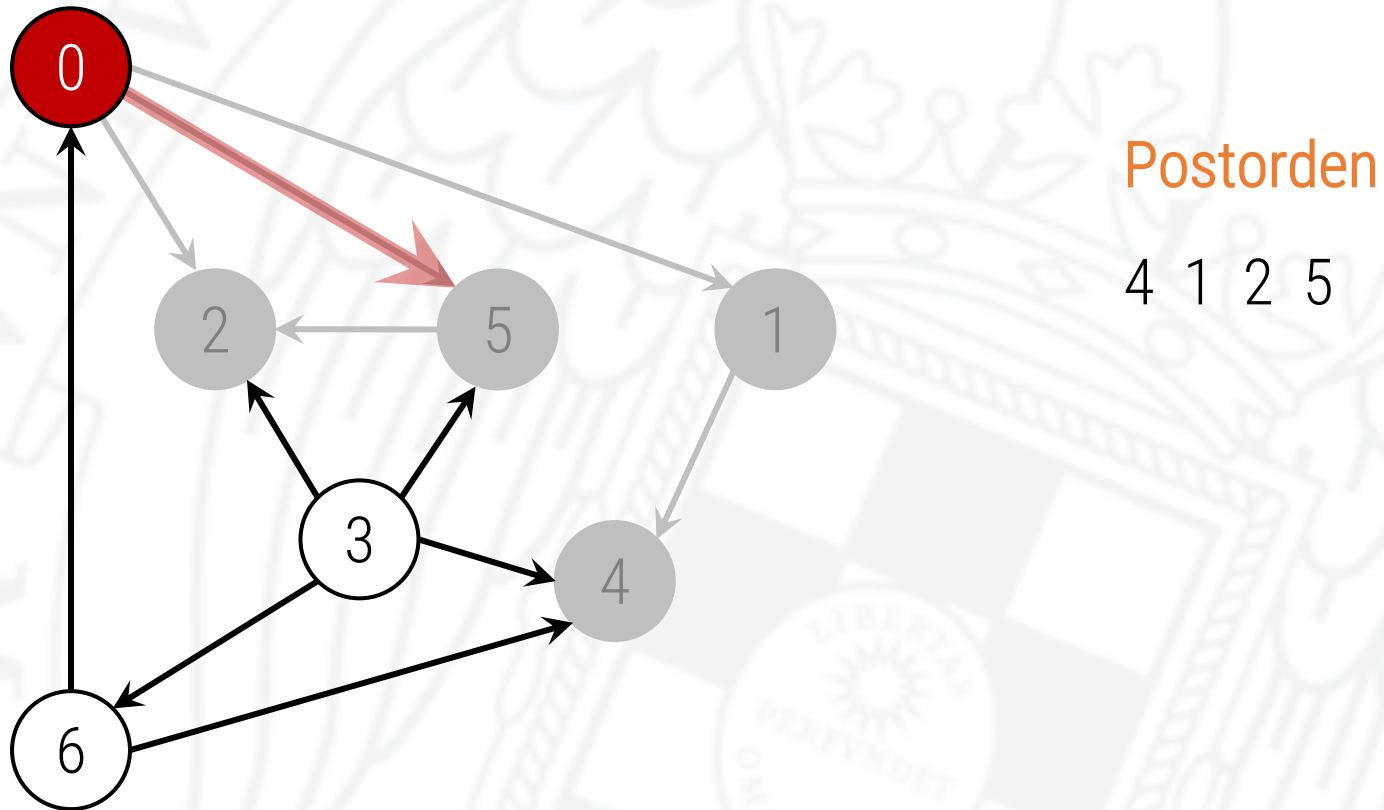
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



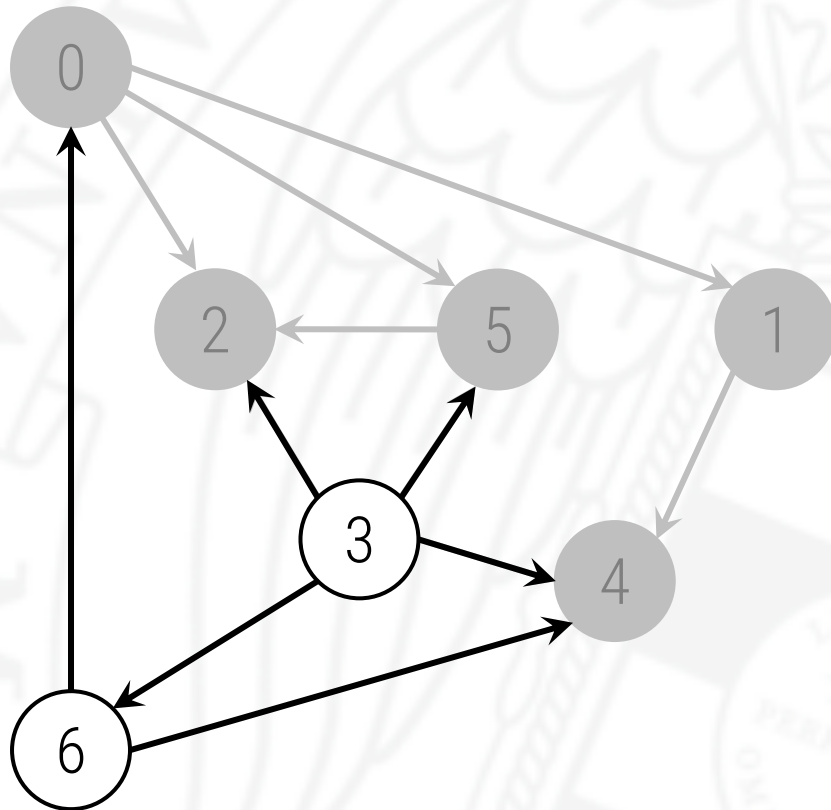
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

- Añadir el vértice al orden tras las llamadas recursivas.

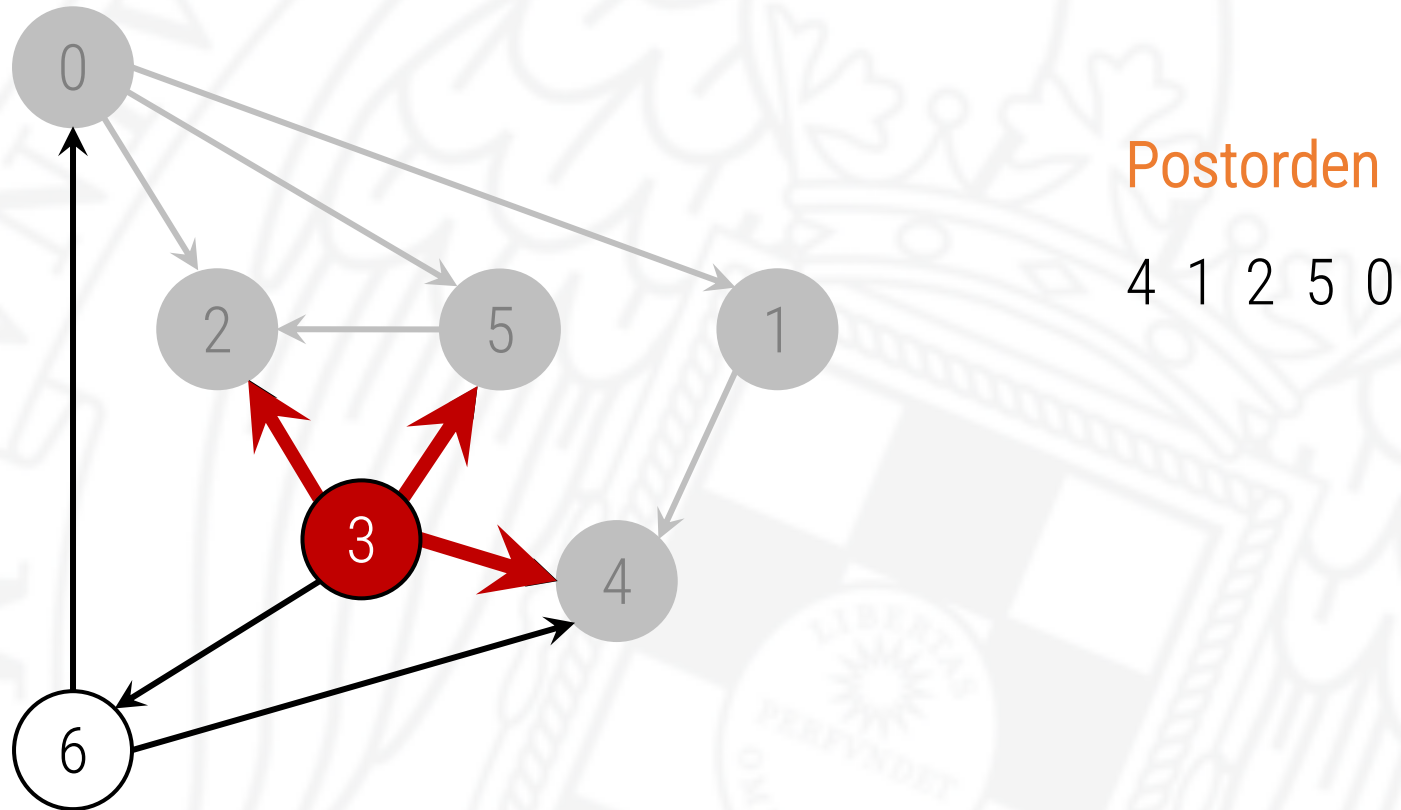


Postorden

4 1 2 5 0

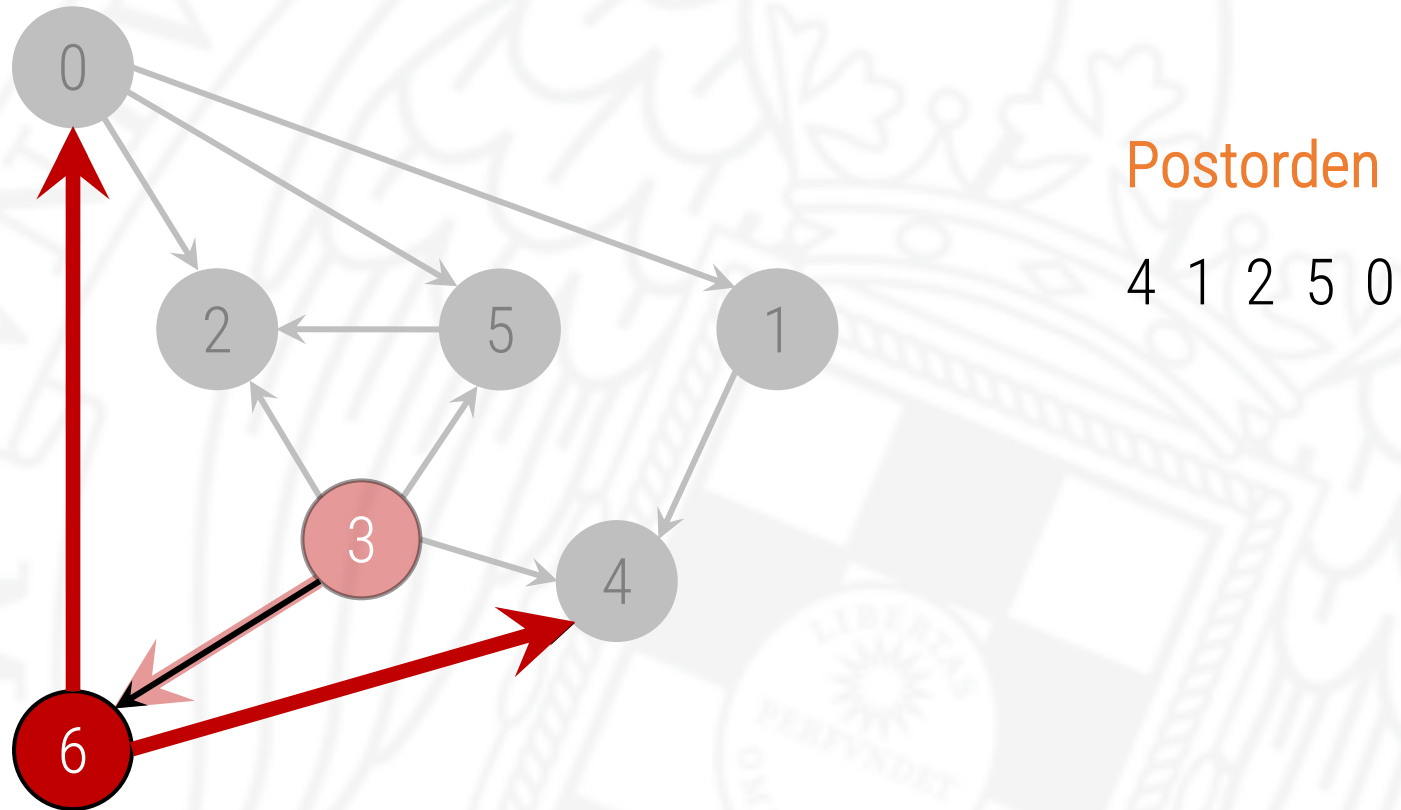
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



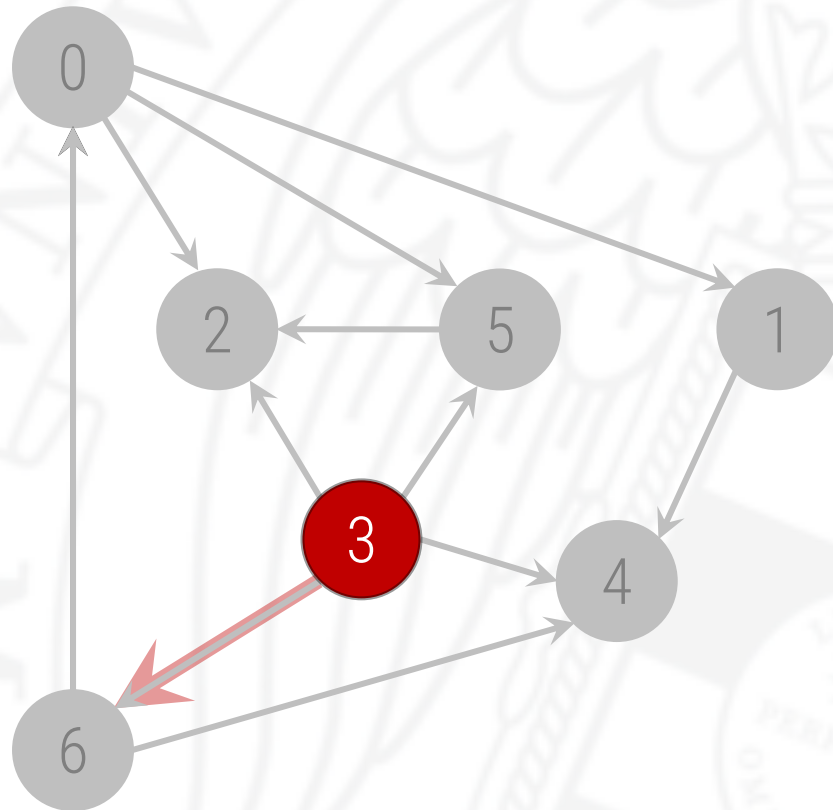
Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

- Añadir el vértice al orden tras las llamadas recursivas.

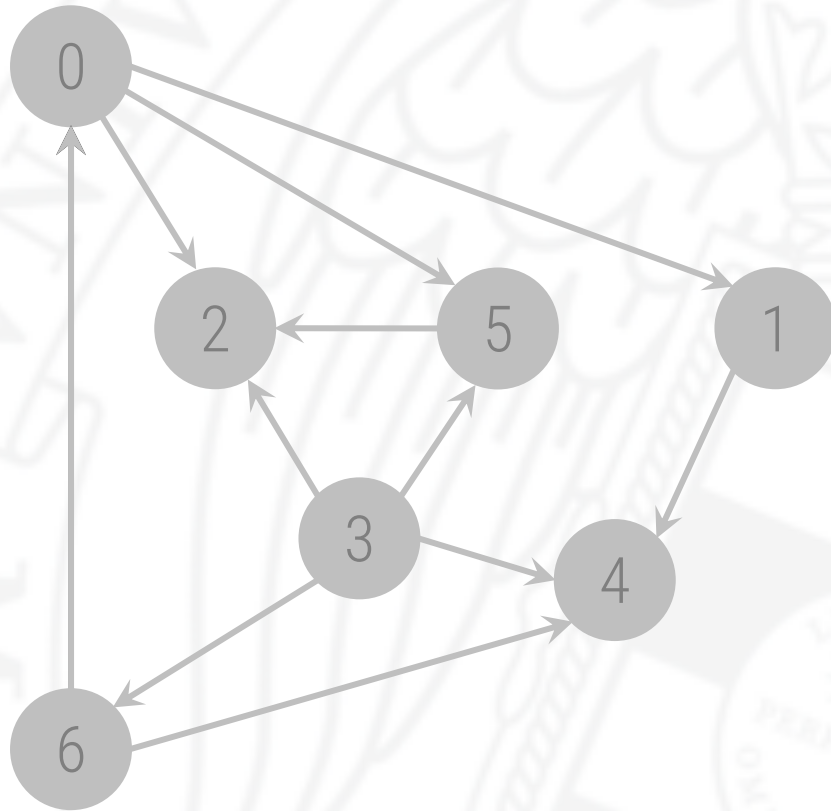


Postorden

4 1 2 5 0 6

Postorden

- Añadir el vértice al orden tras las llamadas recursivas.



Postorden

4 1 2 5 0 6 3

Ordenación topológica (postorden inverso)

3 6 0 5 2 1 4

Corrección

- ▶ El postorden inverso de un DAG es una ordenación topológica.
- ▶ Demostración: Arista cualquiera $v \rightarrow w$. Hacemos la llamada $\text{dfs}(v)$
 - ▶ Caso 1: $\text{dfs}(w)$ ya se hizo y terminó.
 - ▶ Caso 2: $\text{dfs}(w)$ aún no se ha hecho. La arista $v \rightarrow w$ provocará que se haga $\text{dfs}(w)$ y que termine antes de $\text{dfs}(v)$.
 - ▶ Caso 3: $\text{dfs}(w)$ comenzó pero aún no ha terminado. Esto es **imposible**. La cadena de llamadas implica que existe un camino de w a v y la arista $v \rightarrow w$ completaría un ciclo dirigido en un DAG.

Implementación

```
class OrdenTopologico {
public:
    // g es DAG
    OrdenTopologico(Digrafo const& g) : visit(g.V(), false) {
        for (int v = 0; v < g.V(); ++v)
            if (!visit[v])
                dfs(g, v);
    }

    // devuelve la ordenación topológica
    std::deque<int> const& orden() const {
        return _orden;
    }
}
```

Implementación

private:

```
std::vector<bool> visit;
```

```
std::deque<int> _orden; // ordenación topológica
```

```
void dfs(Digrafo const& g, int v) {
```

```
    visit[v] = true;
```

```
    for (int w : g.ady(v))
```

```
        if (!visit[w])
```

```
            dfs(g, w);
```

```
    _orden.push_front(v);
```

```
}
```

```
};
```