

Semana 4. Grafos no dirigidos

1. Conceptos básicos

1.1 Número de aristas en un grafo no dirigido

La fórmula para calcular el número de aristas (E) en un grafo no dirigido dado su número de vértices (n) es:

$$E = \frac{n(n-1)}{2}$$

Esto asume que el grafo es completo, es decir, todos los vértices están conectados entre sí.

1.2 Árbol libre

Un grafo es un **árbol libre** si cumple dos condiciones:

- **Conexo:** No existen vértices aislados, todos los vértices están conectados de alguna manera.
- **Acíclico:** No contiene ciclos, lo que significa que no hay ningún camino en el grafo que empiece y termine en el mismo vértice.

1.3 Grafos isomorfos

Dos grafos son **isomorfos** si tienen la misma estructura, es decir, pueden ser representados de forma idéntica (aunque los vértices pueden tener nombres diferentes). Para comprobar si dos grafos son isomorfos, una estrategia útil es verificar que tengan:

- El mismo número de vértices.
 - El mismo número de vértices con el mismo grado (número de conexiones).
-

2. Representación de grafos

2.1 Matriz de adyacencia

La **matriz de adyacencia** es una forma de representar un grafo usando una matriz $n \times n$, donde n es el número de vértices. Cada elemento de la matriz indica si hay una arista entre el vértice i y el vértice j .

2.2 Lista de adyacentes

La **lista de adyacentes** es una representación más eficiente en términos de espacio, especialmente en grafos dispersos. Consiste en una lista para cada vértice que contiene todos los vértices adyacentes a él.

Esta es la representación más utilizada en la práctica debido a su eficiencia.

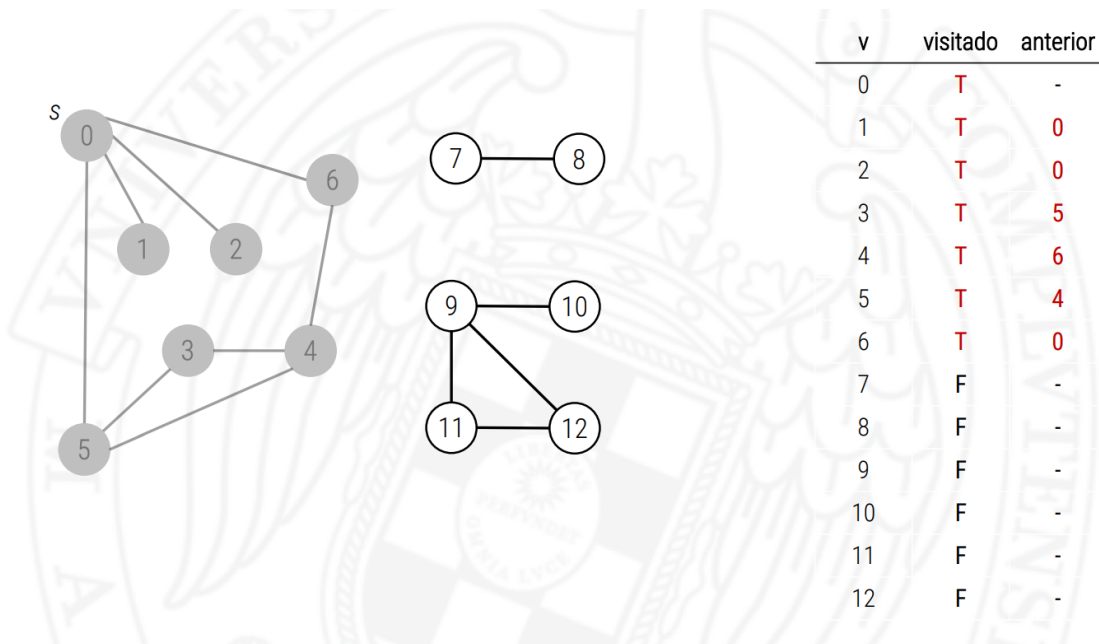
3. Recorridos en grafos

3.1 Recorrido en profundidad (DFS)

Un **recorrido en profundidad** imita el comportamiento de un laberinto, donde se avanza todo lo posible en una dirección antes de retroceder. Los pasos básicos son:

1. **Marcar** el vértice como visitado.
2. **Realizar** alguna acción con el vértice (por ejemplo, imprimir su nombre).
3. **Visitar** recursivamente todos los vértices adyacentes que no hayan sido visitados.

Proceso visual:



Características:

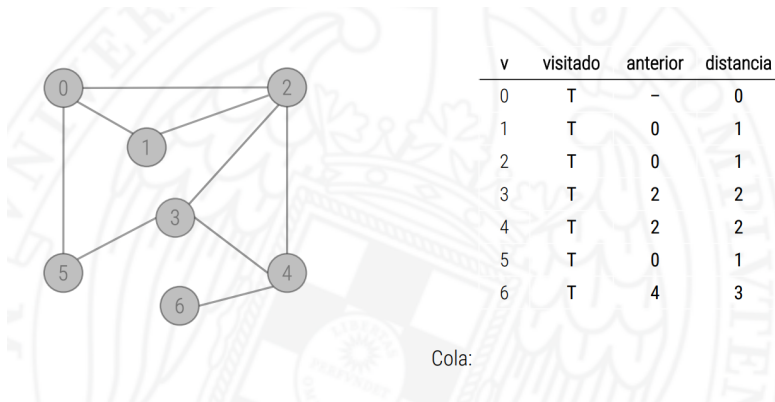
- Visita todos los vértices alcanzables desde el vértice de inicio.
- Vuelve atrás de manera recursiva cuando no puede avanzar más.
- **Complejidad:** $O(A+V)$, donde A es el número de aristas y V el número de vértices.

3.2 Recorrido en anchura (BFS)

El **recorrido en anchura** se utiliza principalmente para encontrar el **camino más corto** desde un vértice origen sss hasta un vértice vvv. El algoritmo sigue estos pasos:

1. Comienza desde un nodo y visita todos sus adyacentes.
2. Mantiene una cola de los vértices por visitar y almacena la **distancia** desde el nodo de origen a cada nodo visitado.
3. La distancia de un nodo s a otro nodo v es la distancia hasta el nodo anterior más uno.

Proceso visual:



Características:

- Parte desde un nodo y visita todos sus adyacentes.
- Almacena la distancia hasta cada nodo y su nodo anterior.
- La distancia de s a un nodo es la distancia de su nodo anterior más uno.
- **Complejidad:** $O(V+A)$, donde V es el número de vértices y A es el número de aristas.