

# AED Tema 6. Grafos I

Martín González Dios

31 de octubre de 2024

## 1. Conceptos y definiciones

- **Grafo**  $G=(V, A)$ : conjunto de vértices o nodos  $V$  y conjunto de arcos o aristas  $A$ .
- **Arco/Arista**  $(u, v)$ : formada por el par de nodos  $u, v$ . Puede ser dirigido (digrafo  $u \rightarrow v$ ) o no  $(u-v)$ .
- **Nodos adyacentes**: nodos relacionados por un arco. Si no están dirigidos  $u$  y  $v$  son adyacentes, pero si  $u \rightarrow v$ ,  $u$  es adyacente a  $v$ , pero  $v$  no es adyacente a  $u$ .
- **Grafo valorado**: sus arcos tienen un factor de peso asociado.
- **Grado de un nodo  $v$** : si el grafo es no dirigido es el número de aristas que contiene a  $v$ . Si el grafo es dirigido se puede calcular el grado de entrada (número de arcos que llegan a  $v$ ) o el de salida (número de arcos que salen de  $v$ ).
- **Camino**: secuencia de vértices  $P=(v_0, v_1, \dots, v_n)/(v_i, v_{i+1}) \in A(\text{arcos})$ . El número de arcos es la longitud.
- **Bucle**: camino  $v-v$ .
- **Camino simple**: camino en el que todos sus nodos son distintos ( $v_0$  puede ser igual a  $v_n$ , los extremos del camino)
- **Ciclo**: camino simple cerrado ( $v_0 = v_n$ ) de por lo menos 3 nodos. Si tiene longitud  $k$  es un  $k$ -ciclo.

## 2. Representación

### 2.1. Representación secuencial con arrays

Se utiliza una **matriz de adyacencia**.

Sea  $G = (V, A)$  un grafo de  $n$  nodos, suponiendo que los nodos  $V = v_1, v_2, \dots, v_n$  están ordenados y que podemos representarlos por sus ordinales  $1, 2, \dots, n$ , la representación de los arcos se hace con una matriz  $A_{n \times n}$  (matriz de adyacencia) definida como  $a_{ij} = 1$  si  $\exists$  arco  $(v_i, v_j)$ ,  $0$  si no  $\exists$  arco  $(v_i, v_j)$ . Si el grafo es no dirigido la matriz será simétrica y si es valorado la matriz será valorada o de pesos.

\*En ocasiones la matriz es booleana con cierto y falso en lugar de  $1$  y  $0$ .

Como la **representación** es **estática** debemos prever el número máximo de nodos. Estos tienen un identificador que se guarda en un vector de cadenas. La matriz de adyacencia es de tipo `int` y el entero  $N$  indica cuantos vértices hay en el grafo. Si el grafo tiene pesos se pueden asociar a la matriz. Lo bueno de esta representación es el orden de **eficiencia de las operaciones de obtención** del coste asociado a un arco, y que la comprobación de conexión entre 2 vértices es independiente del tamaño del grafo. Pero **no se puede modificar el número de vértices**, y es muy disperso (muchos  $0$ s en la matriz  $\rightarrow$  matriz sparse).

## 2.2. Representación dinámica con una estructura multienlazada

Se utiliza una **lista de adyacencia**.

Es una estructura multienlazada formada por una lista directorio cuyos nodos representan los vértices del grafo y listas enlazadas a cada uno de estos nodos que representan los arcos cuyo vértice de origen es el nodo de la lista directorio. Esto **soluciona la ineficacia cuando hay pocos arcos**, usando menos memoria, pero es más complejo y es **ineficiente en la búsqueda de aristas** que llegan a un nodo.

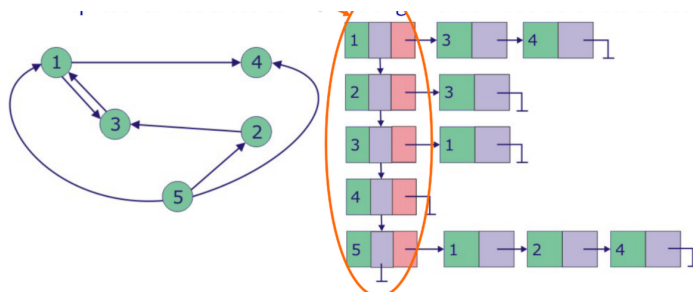


Figura 1: Lista directorio y representación del grafo

## 3. Recorrido de un grafo

Consiste en visitar todos los nodos a partir de un nodo dado. (ver vídeos)

- **Recorrido en anchura no recursivo (estructura auxiliar cola):** visita el vértice de partida V, después visita los adyacentes no visitados y continúa con los adyacentes de estos hasta recorrerlos todos. En la cola se guardan los vértices que se van a ir visitando posteriormente.
- **Recorrido en profundidad recursivo:** visita del vértice de partida V, después recorre en profundidad cada vértice adyacente a V no visitado hasta recorrerlos todos.
- **Recorrido en profundidad no recursivo (estructura auxiliar pila):** sigue la misma lógica que el recorrido anterior, pero va metiendo en una pila los vértices a procesar.

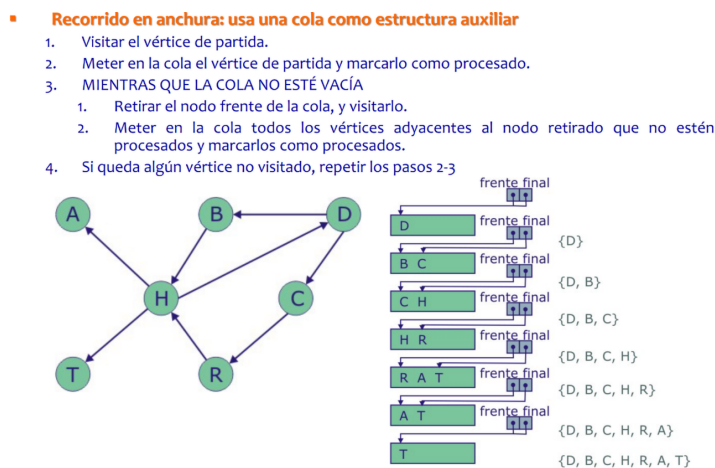


Figura 2: Lista directorio y representación del grafo

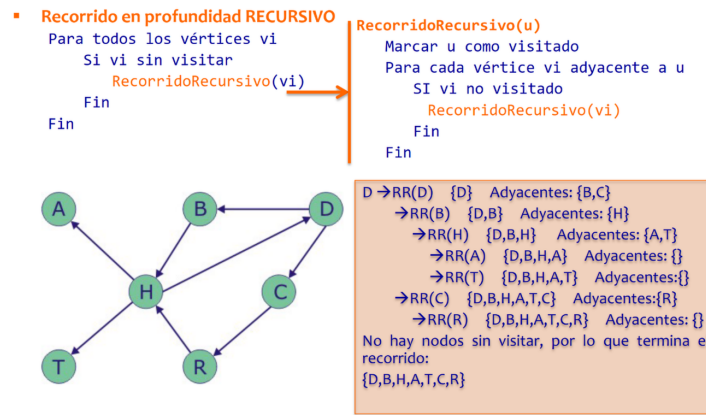


Figura 3: Profundidad recursivo

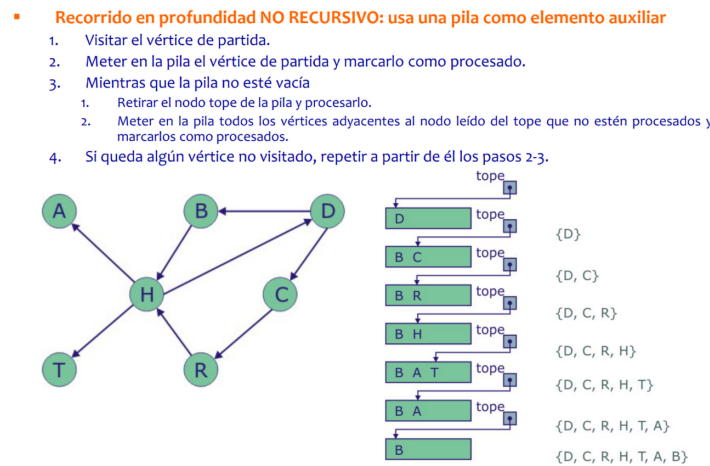


Figura 4: Profundidad no recursivo

## 4. Componentes conexas de un grafo

Un grafo es **conexo** (o fuertemente conexo si es dirigido) cuando existe un camino entre cualquier par de nodos del grafo. En el caso de no ser conexo podemos hablar de **componentes (fuertemente) conexas**.

- **Algoritmo para determinar las componentes conexas de un grafo no dirigido:** Se recorre el grafo a partir de cualquier vértice y los vértices visitados se guardan en el conjunto  $W$ . Si este contiene todos los vértices, el grafo es conexo, si no  $W$  es un componente conexo, y se repite el proceso realizando el recorrido desde otro nodo hasta visitar todos los nodos.
- **Algoritmo para determinar los componentes conexas de un grafo dirigido:** se obtienen los **descendientes de un vértice de partida  $v$** , este incluido, con un recorrido en profundidad, recogiendo todos los vértices a los que les llega un camino desde  $v$ . Después se obtienen los **ascendentes de  $v$** , este incluido, invirtiendo las direcciones del grafo y haciendo lo anterior para obtener los vértices de los que sale un camino que llega a  $v$ . La **unión de ambos conjuntos** es el conjunto de vértices del componente fuertemente conexo al que pertenece  $v$ . Si es igual al grafo este será fuertemente conexo, si no, se seleccionará otro vértice que no pertenezca a este conjunto y se repetirá el proceso hasta encontrar todos los componentes fuertemente conexas del grafo.

## 5. Matriz de caminos

Sea  $A$  una matriz de adyacencia,  $A^m$  es una matriz que indica los caminos de grado  $m$  que existen entre 2 vértices. La matriz de caminos  $\mathbf{P}$  tendrá **1** si hay un camino de  $v_i$  a  $v_j$  (equivalente a que el elemento de la matriz  $B = A + A + \dots + A^n$  sea  $\geq 1$ ) y un **0** si no.

Si un grafo es fuertemente conexo, la matriz de caminos tendrá 1 en todos los elementos excepto en la diagonal (para todos los vértices  $v_i$ ,  $v_j$  existe un camino en ambos sentidos).

## 6. Punto de articulación de un grafo

Es un vértice que al eliminarlo junto con sus arcos el componente conexo en el que se encuentra se divide en dos o más componentes.

Se realiza un **recorrido en profundidad recursivo** del grafo, que se puede representar con un árbol de recubrimiento. Cada arco del grafo será una arista del árbol, produciendo aristas hacia atrás si un vértice de los que une el arco ya estaba visitado anteriormente.

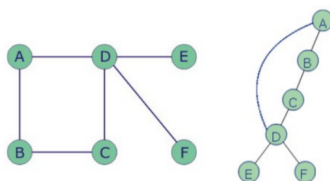


Figura 5: Árbol de recubrimiento

Se realizan los siguientes pasos:

- Se numeran los nodos en el orden en el que se visitaron en el recorrido en profundidad. A esta numeración se le llama **Num(v)**.
- Para cada vértice del árbol determinamos **Bajo(v)**, que es el número mínimo entre 3 números:
  - $\text{Num}(v)$ .
  - El menor valor de  $\text{Num}(w)$  para los vértices  $w$  de las aristas hacia atrás del vértice  $v$ .
  - El menor valor de  $\text{Bajo}(w)$  para sus descendientes  $w$ .
- Determinamos los puntos de articulación: la raíz es punto de articulación si y sólo si tiene 2 hijos. Cualquier vértice  $w$  es punto de articulación si y sólo si tiene al menos un hijo  $u$  tal que  $\text{Num}(w) \leq \text{Bajo}(u)$ .