

# Video36 (1)

---

## Conceptos Principales

---

1. a
2. b
3. c

## Notas

---

### Resumen del video: Recorridos en Grafos (DFS y BFS)

En grafos, a veces no queremos hacer nada súper avanzado; solo **recorrer todo el grafo** para imprimirla, verificar conexiones o preparar otros algoritmos. Para eso existen dos recorridos clave:

#### **1. Recorrido en Profundidad (DFS) 2. Recorrido en Anchura (BFS)**

Los dos hacen lo mismo (visitar todos los nodos), pero **el orden cambia completamente** porque usan diferentes estructuras de apoyo.

---

## Representación del grafo: ¿lista o matriz de adyacencia?

---

El profe hace una pausa para recordar algo importante:

- **Matriz de adyacencia:**

Fácil de trabajar, pero ocupa mucha memoria si hay pocos enlaces.

Gran parte de la matriz serían ceros.

- **Lista de adyacencia:**

Mucho más eficiente cuando el grafo tiene pocas aristas.

No desperdicia memoria.

### **En exámenes:**

Matriz = buena si el grafo es denso.

Lista = mejor si es disperso.

En el video usan matriz, pero el profe dice que para entender algoritmos es mejor guiarnos por el dibujo del grafo.

---

# 1. Recorrido en Profundidad (DFS)

---

## Idea principal:

Avanza por un camino hasta que ya no pueda más.

Como *meter la mano en una bolsa hasta el fondo*.

## Estructura de apoyo:

**Una pila (stack)** → porque actúa último en entrar, primero en salir.

Pasos del algoritmo (siempre son estos):

1. Elegir nodo inicial (en el video usan A).
2. Meterlo a la pila.
3. Repetir:
  - Hacer **pop** de la pila → “procesarlo”.
  - Marcarlo como procesado.
  - Insertar todos sus **adyacentes NO visitados** en la pila.

El detalle importante es que **DFS marca nodos en el orden en el que se hunde en el grafo**, siguiendo un camino lo más profundo posible antes de regresar.

¿Qué problema resuelve?

- Encontrar caminos profundos.
- Explorar desde un punto hacia “abajo”.
- Útil para detectar ciclos, encontrar componentes, hacer backtracking, etc.

Cómo se ve en el recorrido:

El DFS del video va siguiendo rutas largas, por ejemplo:

A  
→ M  
→ F  
→ E  
→ J  
→ H  
→ C  
→ G  
→ ...

Se nota que "se mete" por uno de los lados del grafo antes de volver a subir.

---

## 2. Recorrido en Anchura (BFS)

---

### Idea principal:

Visita el grafo *por niveles*, como ondas que se expanden desde un punto inicial.

Primero los vecinos cercanos, luego los vecinos de esos, y así.

### Estructura de apoyo:

**Una cola (queue)** → primero en entrar, primero en salir.

Pasos del algoritmo:

1. Elegir nodo inicial (también A).
2. Meterlo en la cola.
3. Repetir:
  - Sacar el primero de la cola (dequeue).
  - Procesarlo y marcarlo.
  - Insertar en la cola sus adyacentes no visitados.

¿Qué problema resuelve?

- Buscar el camino más corto en grafos sin pesos.
- Recorrer por niveles.
- Útil para búsquedas globales antes de ir profundo.

Cómo se ve en el recorrido:

BFS imprime primero:

A

Luego todos los vecinos directos:

B, F, M...

Después los que están "un nivel más lejos":

H, J, Y, etc.

Y así hacia afuera en "capas".

En el dibujo se nota como si el recorrido creciera "en círculos".

---

Diferencia clave entre DFS y BFS (la que sí va para examen)

**DFS** = se mete profundo antes de explorar otras opciones.

→ Usa **pila**.

→ Sigue una línea larga.

**BFS** = explora por niveles, como ondas.

→ Usa **cola**.

→ Expande en círculos.

Y por eso el orden de visita cambia totalmente aunque el grafo sea el mismo.