



# UT7 Grafos Dirigidos

## Conceptos Básicos

- **Grafos:** Modelos naturales para representar relaciones entre objetos de datos.
  - **Componentes:**
    - **Vértices (V):** Elementos o nodos del grafo.
    - **Arcos (A):** Conexiones entre pares de vértices.
  - **Tipos de Grafos:**
    - **No dirigido:** Arcos sin dirección específica, es decir,  $(v_i, v_j) = (v_j, v_i)$ .  
 $(v_i, v_j) = (v_j, v_i) \quad (v_i, v_j) = (v_j, v_i)$
    - **Dirigido:** Arcos con dirección, representados por pares ordenados de vértices.

## Propiedades de los Grafos

- Existe como máximo una arista conectando cualesquiera dos vértices.
- Dos vértices son adyacentes si hay una arista que los conecta.
- Un grafo está conectado si existe un camino entre cualquier par de vértices.

## Representaciones de Grafos Dirigidos

- **Matriz de Adyacencias:** Utiliza un espacio del orden de  $n^2$ , donde  $n$  es el número de vértices.  
 $n^2$   
 $n$
- **Lista de Adyacencias:** Requiere espacio proporcional a la suma de la cantidad de arcos y vértices, y puede ser estática o dinámica.

## Ejemplos de Uso de Grafos

- **Ciudades y distancias:** Los vértices representan ciudades y los arcos las distancias entre ellas.
- **Programas de computador:** Los vértices representan bloques de un programa y los arcos las transferencias de flujo de control.
- **Asignaturas universitarias:** Los vértices representan asignaturas y los arcos las relaciones de prerequisites entre ellas.
- **Autómatas:** Los vértices representan estados y los arcos las transiciones.
- **Gestión de proyectos:** Los vértices representan eventos de inicio y fin de tareas y los arcos las tareas necesarias para ejecutar un proyecto.

## Caminos en Grafos Dirigidos

- **Camino:** Secuencia de vértices donde cada par consecutivo está conectado por un arco.
  - **Longitud del camino:** Número de arcos en el camino.
  - **Camino simple:** Todos sus vértices son distintos, excepto tal vez el primero y el último.
  - **Ciclo:** Camino simple que empieza y termina en el mismo vértice y tiene al menos dos arcos.

## Algoritmos para Grafos Dirigidos

- **Algoritmo de Dijkstra:**
    - Resuelve el problema de los caminos más cortos desde un vértice origen a todos los demás vértices.
    - Utiliza una técnica "ávida" (greedy) para construir un conjunto de vértices con la distancia más corta conocida al origen.
    - **Pasos:**
      - Inicializa el conjunto de vértices S con solo el origen.
- SS

- En cada paso, se agrega el vértice restante con la distancia más corta desde el origen.
- Actualiza el vector D que registra la longitud del camino más corto a cada vértice.

DD

- **Algoritmos Ávidos:**

- Seleccionan iterativamente el mejor candidato posible.
- **Elementos:**
  - Conjunto de candidatos.
  - Función solución.
  - Función de selección.
  - Función de factibilidad.
  - Función objetivo.
- **Proceso:**
  - Elegir el mejor elemento posible del conjunto de candidatos.
  - Comprobar si produce una solución factible.
  - Incluirlo en el conjunto solución si es factible.
  - Repetir hasta alcanzar la función objetivo o agotar los candidatos.

## Problema de Caminos Más Cortos con un Origen

- **Objetivo:** Determinar el costo del camino más corto desde el origen a cada uno de los demás vértices.
- **Técnica:** "Ávida" (greedy).
  - **Inicialización:** Conjunto S con el origen, vector D con las distancias iniciales.

SS

DD

- **Proceso:** Iterativamente, se agrega el vértice con la distancia más corta y se actualizan las distancias.

## Recuperación de Caminos

- Usar un array P para registrar el vértice inmediato anterior en el camino más corto.

PP

- **Proceso de recuperación:**
  - Para encontrar el camino más corto, se recorren los predecesores en orden inverso.

## Problema de Caminos Más Cortos entre Todos los Pares

- **Objetivo:** Obtener una tabla que indique el menor camino entre todos los pares de vértices.
- **Aplicación:** Ejemplo de tiempos de vuelos entre ciudades.