

# Ejercicio 11 TDA-P3

Aguilar valentin

May 2, 2024

Análisis de Implementaciones de Vecindarios en Grafos

## 0.1 a) Inicializar la estructura a partir de un conjunto de aristas de $G$

- Ventajas:  $O(m)$ .

## 0.2 b) Determinar si dos vértices $v$ y $w$ son adyacentes

- Lista de aristas:  $O(m)$ .
- Lista de adyacencias:  $O(d(v))$ .
- Matriz de adyacencias:  $O(1)$ .

## 0.3 c) Recorrer y/o procesar el vecindario $N(v)$ de un vértice $v$ dado

- Lista de aristas:  $O(m)$ .
- Lista de adyacencias:  $O(d(v))$ .
- Matriz de adyacencias:  $O(n)$ .

## 0.4 d) Insertar un vértice $v$ con su conjunto de vecinos $N(v)$

- Lista de aristas:  $O(1)$ .
- Lista de adyacencias:  $O(n)$ .
- Matriz de adyacencias:  $O(n^2)$ .

### 0.5 e) Insertar una arista $vw$

- Lista de aristas:  $O(m)$ .
- Lista de adyacencias:  $O(d(u) + d(v))$ .
- Matriz de adyacencias:  $O(1)$ .

## 1 Análisis de Estructuras de Datos

### 1.1 a) Lista de aristas

- Ventajas:
  - Puedo acceder a vecindarios en  $O(1)$ .
  - Puedo agregar y sacar vértices en tiempo lineal.
  - Puedo agregar aristas en tiempo decente.
- Desventajas:
  - A diferencia de la matriz, no accedo en  $O(1)$  a las adyacencias (la complejidad es más lenta).

### 1.2 b) Lista de adyacencias pero cada vecino se almacena junto con un índice a la posición que ocupa en la lista de adyacencias del otro vértice

- Ventajas:
  - Acceso más rápido para operaciones bidireccionales.
  - Simplificación de operaciones de eliminación.

### 1.3 c) Matriz de adyacencias

- \* Ventajas:
  - Acceso en  $O(1)$  a las adyacencias.
- \* Desventajas:
  - Ocupa  $O(n^2)$  en memoria.
  - Complejidad de  $O(n^2)$  para borrar y agregar vértices.

### 1.4 d) Mix entre lista de adyacencias y matriz de adyacencias

- \* Ventajas:
  - Acceso rápido para comprobar adyacencias ( $O(1)$ ).
  - Inserciones y eliminaciones rápidas.

- Uso eficiente de memoria.
- \* Desventajas:
  - Sobrecarga de memoria para las tablas de hash.
  - Menos predictibilidad en los tiempos de ejecución.