Bases de datos

Clase 2

Grafos

Un grafo puede definirse como G = (V, A), donde V representa a un conjunto de puntos, llamados vértices o nodos, y A es un conjunto de relaciones entre pares de vértices, llamadas aristas o arcos. De esta forma un grafo es un conjunto de vértices y arcos que los relacionan.

Ahora, las aristas se van a llamar **Relaciones** y los vértices se llaman **nodos** porque contienen información

**Grado ->** Es la cantidad de arcos o relaciones que entran o salen de un nodo

* Grado positivo -> la cantidad de arcos que salen de un nodo
* Grado negativo -> la cantidad de arcos que entran en mi nodo

Si el Grafo es direccionado y me piden el grado, elijo el mas grande entre el positivo y el negativo

Los grafos son una forma de modelar los problemas tal que los nodos son los elementos del problema y las relaciones son las relaciones entre dichos elementos.

Los grafos son abstracto -> no existen en realidad, son un concepto y tienen una **representación computacional.** Esta representación depende del tamaño que pueda tomar, si es estático, probablemente no cambie su tamaño, si es dinámico, si.

* Representación estática -> Si bien contempla los cambios en mi grafo, no cambia su estructura cada vez que hay un cambio en mi grafo. Esta pensada para ser modificada muy de vez en cuando
  + Matriz de adyacencia-> matriz de nxm donde n es la cantidad de nodos y hay un 1 si los nodos están relacionados y 0 si no

A diagram of a triangle with points and lines

AI-generated content may be incorrect.

* + Matriz de incidencia -> matriz de nxm donde n es la cantidad de nodos y m la cantidad de relaciones

A diagram of a triangle and a square

AI-generated content may be incorrect.

* Representación dinámica -> Cambia cada vez que se produce un cambio en mi grafo, es decir, que se agreguen o saquen relaciones y/o nodos
  + Lista de adyacencia o estructura de graal -> Lista de nodos y cada uno de esos nodos tiene un puntero a una lista que tiene las relaciones. Estas se representan en la dirección de cada nodo, es decir, si el nodo ‘a’ se relaciona con el ‘b’, el nodo ‘a’ tiene una lista con todas las direcciones de los nodos con los que se relaciona, en la que se encuentra, la dirección del nodo ‘b’

Características de grafos

* Grafo libre -> grado 0, no tiene relaciones, están todos los nodos sueltos
* Grafo completo -> tiene todas las relaciones posibles entre nodos
* Grafo regular -> todos los nodos tienen el mismo grado
* Grafo simple -> como mucho una arista une 2 vertices para todos los vértices
* Grafo complejo -> mas de una arista, une dos vértices
* Grafo conexo -> para cualquier par de vértices hay un camino que puedo seguir para llegar de uno a otro
* Grafo no conexo -> hay un grupo de vértices que no está conectado con el resto de vértices
* Grafo complementario a otro grafo -> grafo que tiene las relaciones que le faltan al otro.

Clasificación de grafos

* Dirigidos -> relaciones unidireccionales entre nodos, van en un sentido
* No dirigidos -> relaciones bidireccionales entre nodos
* Grafos restrictos -> Los grafos **no deben cumplir** con las características de reflexividad, simetría y transitividad, es decir, es equivalente
  + Reflexividad -> nodo se relaciona con si mismo
  + Simétrico -> si a se relaciona con b, b se relaciona con a
  + Transitividad -> si ***a*** se relaciona con ***b*** y ***b*** se relaciona con ***c***, entonces a se relaciona con ***c.*** Llevar a un esquema directo una relación indirecta.
    - Anti -> nunca se cumple la característica en ninguna relación
    - a- -> puede o no cumplirse la característica en la relacion
* Grafos irrestrictos -> pueden o no cumplir con las características previamente mencionadas

Camino paso y ciclo en grafo

* Paso -> solo en grafos dirigido. Hay un camino entre 2 nodos siguiendo el sentido de la relación.
* Camino -> hay una vinculación directa o indirecta entre 2 o mas arcos independientemente del sentido de los arcos
  + Si hay camino hay paso pero no necesariamente al revés
* Ciclo -> camino que empieza y termina en el mismo nodo

Búsqueda de camino en grafo

* Búsqueda en profundidad (Depth first): avanzo en profundidad, es decir, sin mantener un orden jerárquico de evaluación, donde la técnica avanza hasta que no pueda avanzar mas y de ahí vuelve para atrás. No analiza todos los arcos que salen de un nodo, va al destino del primer arco de un nodo
* Búsqueda en anchura (breath first): busca en anchura, es decir, busca en todos los nodos relacionados con 1 y si ninguno es, pasa a el primero de ellos y hace lo mismo, busca en todos los relacionados con este.

Estructura de datos

Una estructura de datos es un **grafo dirigido y restricto**, con las características de unicidad en sus relaciones, esto es que en orden de predecesor, cada nodo solo puede tener un nodo predecesor a él. Se usan para modelar problemas reales como los grafos que, como son restrictos, son mas administrables.

Estas pueden ser univocas y biunívocas:

* Univoco -> tiene un solo predecesor pero muchos posteriores
* Biunivoco -> tiene un predecesor y un solo posterior. Cualquier elemento del dominio de entrada, tiene un solo elemento en el dominio de salida diferente. Pila, cola, lista.

Listas-> puede estar linkeada o ser estatica. Los elementos pueden ser posicionados donde necesite y/o ordenados según necesite. Tipos de listas:

* Lineal -> empieza y el ultimo nodo termina en null
* Circular -> empieza y el ultimo nodo apunta al primero
* Lista doblemente enlazada cada nodo tiene 2 punteros, donde uno hace referencia al siguiente y otro que apunta al anterior.

Pila -> puedo representarla dinámicamente (con punteros) o estáticamente (con vectores). Esta tiene un tope y todos entran y salen por el mismo tope (LIFO)

Cola -> tiene 2 punteros, al primero que entro y al ultimo que entro. El primer elemento en salir es el primero en entrar. Puedo representarlo dinámicamente o estáticamente

Aboles -> el grado define la cantidad de punteros que tiene cada nodo. Funciona como una lista, pero tiene mas de una rama en su representación. Estáticamente es un vector