Teoría de Algoritmia. 1º de Carrera. Tema 1

Índice

1 Introducción

Un grafo combinatorio es una pareja ordenada $G = (V, E)$ de V vértices y un subconjunto E contenido en $V \times V$ siendo este el producto Cartesiano $V \times V$. Si el grafo no está dirigido se llaman aristas y las parejas $(v, w) \in E$ se consideran sin orden. Si el grafo está dirigido u orientado los elementos de E se llaman flechas, las cuales poseen una dirección, y las parejas (v, w) un orden. Si el grafo no está dirigido, no se considera que tenga un principio o un final, simplemente están relacionados.

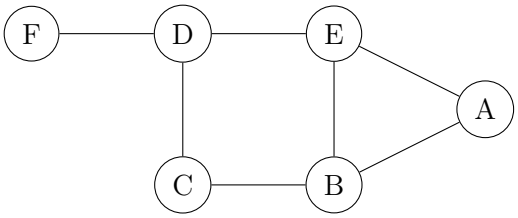


Figure 1: Grafo No Dirigido

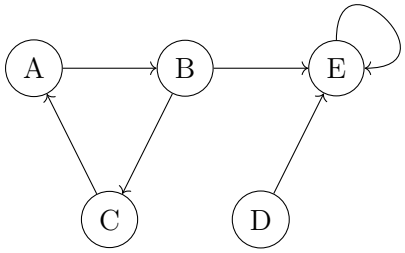


Figure 2: Grafo Dirigido

Los grafos se usan para representar redes de comunicación, organizaciones de datos y flujos de computación. La primera persona que habló de grafos fue Leonhard Euler resolviendo el problema de los siete puentes de Königsberg. En este problema surgió lo que hoy es conocido como la **Fórmula de Euler**, la cual relaciona el número de aristas, vértices y caras de un poliedro convéxo.

Uno de los problemas más clásicos en teoría de grafos fue el problema de los cuatro colores:

¿Es cierto que cualquier mapa dibujado en el plano puede colorearse con 4 colores de tal forma que cualquieras dos regiones que tengan una frontera común tengan diferentes colores?

2 Definiciones Básicas

2.1 Orden y medida

El Orden de un grafo es el número de vértices de un grafo ($|V|$). La medida de un grafo es el número de aristas o flechas ($|E|$)

2.2 Valencia y Grado

El grado o valencia de un vértice es el número de aristas llegando o saliendo de él. Si una arista une un vértice consigo mismo cuenta 2 veces.

El in-grado (*in-degree originalmente*) es el número de aristas que llegan al vértices. El out-grado (*out-degree originalmente*) es el número de aristas saliendo del vértice. Estos valores solo se pueden cálcular en un grafo dirigido.

2.3 Vértice Hoja y Vértice Rama

Un vértice hoja, es un vértice que posee una única arista. También pueden ser llamados terminales.
Un vértice rama, es aquel vértice que está conectado por más de dos aristas.

NOTA: *Si un vértice está unido por dos aristas, una a otro vértice y otra a si mismo, en un grafo dirigido formalmente se considera rama, pero ya que muchas veces se toma solo la valencia de entrada no sería rama.*

2.4 Camino y Bucles

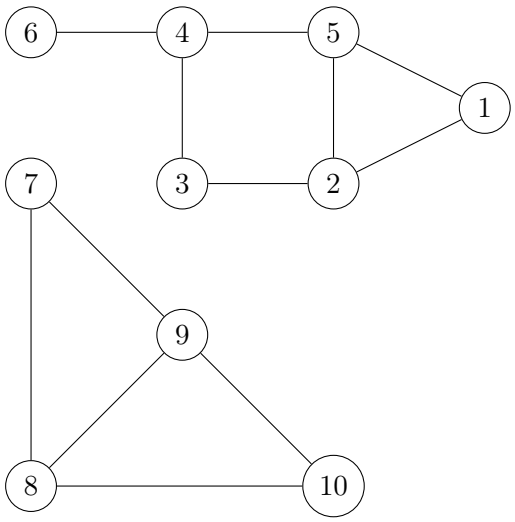
El camino de un grafo es la secuencia de aristas conectadas linealmente. En el grafo dirigido, el final de una arista debe ser el inicio de la siguiente. La longitud de un camino es el número de aristas que contiene.

Un bucle o circuito es un camino cerrado. Eso implica que el final de la última arista es el inicio de la primera.

3 Conectividad de un grafo

3.1 Conectividad de un grafo no dirigido

Un grafo no dirigido está **conexo** si existe un camino entre cualquier pareja de vértices. Un **componente conexo** en un subgrafo en el que todos los puntos están conectados entre sí. En el ejemplo que se muestra a continuación, el grafo es un grafo desconectado con 2 componentes conexos, el primero formado por los nodos 1,2,3,4,5 y el segundo formado por los nodos 7,8,9,10

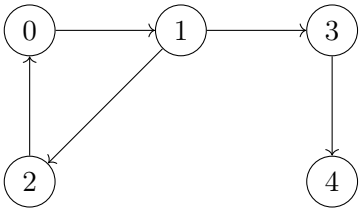


3.2 Conectividad de un grafo dirigido

Se dice que un grafo dirigido está conectado de forma **débil** cuando está conectado como un grafo no dirigido.

Un grafo dirigido esta **semiconectado** cuando para dos vértices cualesquiera (u, v) contiene un camino de u a v **o** de v a u.

Un grafo dirigido está conectado de forma **fuerte** si existen ambos caminos en el grafo semiconectado. Un ejemplo de un grafo dirigido con 3 componentes conectados de forma fuerte es el siguiente:



Los 3 componentes fuertes corresponderían al formado por 0,1 y 2, el segundo formado unicamente por el 3 y el tercero formado unicamente por el 4.

4 Modelos de Memoria

4.1 Lista de Adyacencia

Los vértices estań organizados como estructuras y cada vértice guarda una lista de vértices adyacentes (de ahí el nombre de lista de adyacencia). Esta estructura de datos permite almacenar datos adicionales sobre los vértices.

Una estructura en C se define utilizando el comando *struct*

4.2 Lista de Incidencia

Tanto los vértices como aristas están almacenadas como estructuras. Cada arista guarda los vértices que incide, y de forma opcional los vértices pueden guardar las aristas incidentes. Esto permite almacenar información adicional de tanto vértices como de aristas.

4.3 Matriz de Adyacencia

Es una matriz bidimensional en la que las columnas representan los vértices de inicio y las finals los vértices del final de una arista. La entrada i,j en la matriz, correspondería al número de flechas que empiezan en i y terminan en j. En un grafo no dirigido, esta matriz es simétrica.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Esta matriz representa los datos del grafo dirigido representado en la Figura 2.

4.4 Matriz de Incidencia

Es una matriz bidimensional formada por datos *Bool* (True y False) en el que las filas representan los vértices y las columnas representan las aristas. Sus entradas indican si el vértice de la fila y la arista de la columna son incidentes. Para grafos dirigidos:

- $+1$ indica que el vértice es el origen de la flecha
- -1 indica que el vértice es el final de la flecha

$$\begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

Esta matriz también representan el grafo digido de la Figura 2. Sin embargo, en este caso no se puede representar la flecha que sale del nodo 5 y vuelve a entrar.

5 Árboles y árboels con raiz

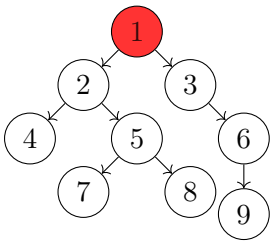
5.1 Árboles

Un **árbol** es un grafo conexo sin bucles. Puede también tomarse como un arbol unicamente arconexo, es decir, para llegar de un nodo a otro del vértice existe un único camino posible. Los árboles presentan una serie de propiedades:

- Si se añade una arista de forma aleatoria se crea un bucle.
- Si se elimina cualquier arista el árbol se vuelve disconexo.
- Un árbol con n vértices tiene exactamente $n-1$ (La caracterísitca de Euler es igual a 1)

5.2 Árboles con raiz

Un árbol con raiz es un árbol en el que un vértice ha sido definido como **raiz** o **nodo base**. Un ejemplo de un árbol con raiz podría ser el siguiente:



En un árbol con raiz, los vértices hoja y rama son vértices hoja y rama comunes excepto el vértice raiz. Es importante tener en cuenta que la valencia de estos vértices no depende del nodo raiz, es decir, son indpendientes de él. La **profundidad** de un vértice se define como la distancia desde el nodo raiz hasta este vértice. La profundidad si depende del nodo raiz. Esta distancia se mide como la distancia del único camino que los une. Por ende, la profundidad de un nodo consigo mismo es 0, luego la profundidad del nodo raiz es 0.

La profundidad de un árbol con raiz se considera la profundidad más larga de los vértices (esto depende de la raiz que se tome).

5.2.1 Nodos padre e hijo

Si tenemos un árbol con raíz y tomamos un nodo v de profundidad $p > 0$, el padre de v es el único nodo conectado a v que presenta una profundidad de $p-1$.Un nodo hijo de u (siendo u un nodo de profundidad $p \geq 0$) es aquel nodo conectado a u que posea una profundidad de $p+1$.

Un árbol n-ario es un árbol con raiz en el que cada nodo presenta n hijos.

5.3 Tree traversal

5.4 Depth-first search: pre-order

Es un algoritmo de búsqueda en el que primero se visita el nodo en el que estés y a continuación el hijo de la izquierda del nodo en el que te encuentres. Si ese nodo tiene hijos se repite este proceso con ese nodo, sino se pasa al siguiente hijo desde la izquierda. El objetivo de esto es que, después de haber visitado el nodo en el que estés, si se puede aumentar la profundidad en la que estás lo hagas, y si no es posible retrocedas hasta poder acceder a un camino nuevo y continuar con este proceso.

5.4.1 Depth-first search: in-order

Este algoritmo primero se mueve hacia el hijo más hacia la izquierda de todos los nodos. Tras esto lo visita, se desplaza hacia su padre directo, lo visita y repite el proceso por la derecha. Si el padre directo no presenta más caminos, se sigue reduciendo la profundidad hasta encontrar un padre con un camino disponible.