Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Gestión de Datos

Teoría de Grafos

Ing. Enrique Reinosa
Julio 2007

Índice

Grafos

Conceptos y Definiciones

Un grafo es una pareja **G = (V, A)**, donde **V** es un conjunto de puntos, llamados ***vértices***, y **A** es un conjunto de pares de vértices, llamadas ***aristas***. Para simplificar, notaremos la arista {a, b} como ab.

En teoría de grafos, sólo queda lo esencial del dibujo: la forma de las aristas no son relevantes, sólo importa a qué vértices están unidas. La posición de los vértices tampoco importa, y se puede variar para obtener un grafo más claro. Generalmente, se considera que colocar los vértices en forma de polígono regular da grafos muy legibles.

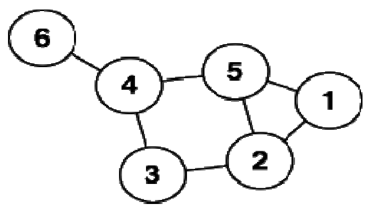
El objetivo de los grafos es la modelización de problemas simples o complejos, como ejemplo, cualquier red puede ser modelada con un grafo: una red de autopistas que conectas ciudades, una red eléctrica o una de subterráneos de una ciudad.

En la figura, Vértices = { a, b, c, d, e, f }, y Arcos = { ab, ac, ae, bc, bd, df, ef }.

Los grafos son colecciones de objetos llamados ***vértices*** o ***nodos***, conectados por líneas denominadas ***aristas*** o ***arcos***. Un grafo es utilizado, en la matemática y en las ciencias de la computación, para representar relaciones entre diferentes elementos, de esta forma los elementos son los nodos y la relación se manifiesta en los arcos que vinculan a dichos nodos entre si. Esas relaciones pueden mantener o no jerarquía, en caso que se presente la necesidad de soportar una relación jerárquica, el grafo debe incluir en sus arcos un sentido de dirección de la relación, el cual marca el orden de evaluación de la relación, si por el contrario la relación no requiere un orden específico los arcos no tienen sentido, como por ejemplo en la relación "=".

En la concepción matemática, la importancia de un grafo radica en establecer relaciones entre los vértices y las aristas. En las ciencias de la computación, si bien es importante el análisis de las propiedades de estas relaciones, es también muy relevante la figura de los vértices, es por ello que la terminología utilizada para denotar los vértices es nodos, esto es porque se los considera como un conjunto de datos, que puede estar formado por un solo elemente como ser un número, una letra, un string, o por muchos de ellos y hasta otro grafo completo puede llegar a ser un nodo de un grafo específico. Esta característica potencia aún más la importancia de los grafos en la modelización de datos.

El siguiente dibujo muestra un grafo donde los círculos representan los nodos y las líneas los arcos que los relacionan.



En algunos casos es necesario asignar un sentido a las aristas, por ejemplo, si se quiere representar la red de las calles de una ciudad con sus inevitables direcciones únicas. El conjunto de aristas será ahora un subconjunto de todos los posibles pares ordenados de vértices, con (a, b) ≠ (b, a). Los grafos que contienen aristas dirigidas se denominan **grafos dirigidos** u **orientados**, como el siguiente:

Aquí Vértices = { a, b, c, d, e }, y A = { (a, c), (d, a), (d, e), (a, e), (b, e), (c, a), (c, c), (d, b) }.

Si las aristas no son orientadas se consideran bidireccionales a efectos prácticos equivale a decir que existen dos aristas orientadas entre los nodos, cada una en un sentido.

En el grafo anterior se ha utilizado una arista que tiene sus dos extremos idénticos: es un lazo o bucle, y aparece también una arista bidireccional, y corresponde a dos aristas orientadas.

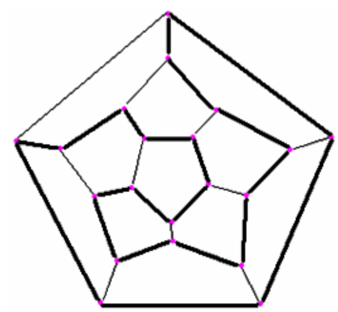
Se considera la característica de "grado" (positivo o negativo) de un vértice ***v*** y se indica como **(*v*)**, a la cantidad de aristas que llegan o salen de él; para el caso de grafos no orientados, el grado de un vértice es simplemente la cantidad de aristas que tocan este vértice. Por ejemplo, el grado positivo (salidas) de *d* es 3, mientras que el grado negativo (llegadas) de *b* es 1.

Caminos, pasos y ciclos

Un **camino** entre dos nodos *a* y *b*, se produce cuando existe una vinculación directa o indirecta entre ambos, esto es cuando puedo vincular mediante uno o mas arcos a dichos nodos entre sí. Dado que en este concepto no está vinculado con la dirección de los arcos, puede existir camino tanto en grafos dirigidos como en grafos no dirigidos.

Un **paso** entre dos nodos *a* y *b*, se produce cuando existe un camino entre ambos, pero con un sentido preestablecido, esto es que partiendo del nodo *a* y siguiendo el sentido de los arcos se llega al nodo *b*. Como en este caso es relevante el sentido, solo se evalúan pasos en los grafos dirigidos.

Un **ciclo** es un camino, es decir una sucesión de aristas adyacentes, donde no se recorre dos veces la misma arista, y donde se regresa al punto inicial. Un **ciclo hamiltoniano** tiene además que recorrer todos los vértices exactamente una vez (excepto el vértice del que parte y al cual llega).



Ejemplo de un ciclo hamiltoniano.

Por ejemplo, en un museo grande (como por ejemplo el Louvre), lo idóneo sería recorrer todas las salas una sola vez, esto es buscar un ciclo hamiltoniano en el grafo que representa el museo (los vértices son las salas, y las aristas los corredores o puertas entre ellas).

Se habla también de camino hamiltoniano si no se impone regresar al punto de partida, como en un museo con una única puerta de entrada. Por ejemplo, un caballo puede recorrer todas las casillas de un tablero de ajedrez sin pasar dos veces por la misma: es un camino hamiltoniano. Ejemplo de un ciclo hamiltoniano en el grafo del dodecaedro.

Hoy en día, no se conocen métodos generales para hallar un ciclo hamiltoniano en tiempo polinómico, siendo la búsqueda por fuerza bruta de todos los posibles caminos u otros métodos excesivamente costosos. Existen, sin embargo, métodos para descartar la existencia de ciclos o caminos hamiltonianos en grafos pequeños.

Como se desprende de los conceptos vertidos, en un grafo pueden existir muchos caminos, pasos o ciclos, entre dos nodos, esto genera la necesidad de medir las diferentes longitudes de cada uno. De esta forma vamos a definir **longitud de paso** como la cantidad de arcos involucrados.

Caracterización de Grafos

Los grafos pueden caracterizarse según diferentes criterios a continuación, se mostrarán algunos de ellos.

Grafos Simples

Un grafo es ***simple*** si a lo sumo ***sólo*** 1 arista une dos vértices cualesquiera. Esto es equivalente a decir que una arista cualquiera es la única que une dos vértices específicos.

Un grafo que ***no*** es simple se denomina ***complejo***.

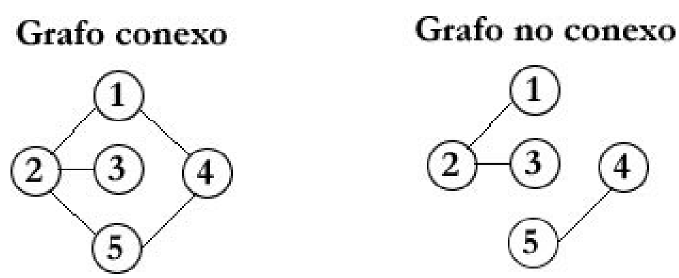
Grafos Conexos

Un grafo es ***conexo*** si cada par de vértices está conectado por un camino; es decir, si para cualquier par de vértices (a, b), existe al menos un camino posible desde *a* hacia *b*.

Un grafo es fuertemente conexo si cada par de vértices está conectado por al menos dos caminos disjuntos; es decir, es conexo y no existe un vértice tal que al sacarlo el grafo resultante sea disconexo.

Es posible determinar si un grafo es conexo usando un algoritmo Búsqueda en anchura (BFS) o Búsqueda en profundidad (DFS).

En términos matemáticos la propiedad de un grafo de ser (fuertemente) conexo permite establecer en base a él una relación de equivalencia para sus vértices, la cual lleva a una partición de éstos en "componentes (fuertemente) conexas", es decir, porciones del grafo, que son (fuertemente) conexas cuando se consideran como grafos aislados. Esta propiedad es importante para muchas demostraciones en teoría de grafos.



Grafos Completos

Un grafo simple es *completo* si existen aristas uniendo *todos* los pares posibles de vértices. Es decir, todo par de vértices (a, b) debe tener una arista *e* que los une.

El conjunto de los grafos completos es denominado usualmente K_1, siendo K_n el grafo completo de *n* vértices.

Un K_n, es decir, grafo completo de *n* vértices tiene exactamente $\frac{n(n-1)}{2}$ aristas.

La representación gráfica de los K_n como los vértices de un polígono regular da cuenta de su peculiar estructura.

Grafos Bipartitos

Un grafo G es bipartito si puede expresarse como $G = \{V_1 \cup V_2, A\}$ (es decir, la unión de dos grupos de vértices), bajo las siguientes condiciones:

- V_1 y V_2 son disjuntos y no vacíos.
- Cada arista de A une un vértice de V_1 con uno de V_2.
- No existen aristas uniendo dos elementos de V_1; análogamente para V_2.

Bajo estas condiciones, el grafo se considera bipartito, y puede describirse informalmente como el grafo que une o relaciona dos conjuntos de elementos diferentes, como aquellos resultantes de los ejercicios y puzzles en los que debe unirse un elemento de la columna A con un elemento de la columna B.

Grafos ponderados

En muchos casos, es preciso atribuir a cada arista un número específico, llamado *valuación*, *ponderación* o *coste* según el contexto, y se obtiene así un **grafo valuado**. Formalmente, es un grafo con una función $v: A \rightarrow \mathbf{R}_+$.

Por ejemplo, un representante comercial tiene que visitar *n* ciudades conectadas entre sí por carreteras; su interés previsible será minimizar la distancia recorrida (o el tiempo, si se pueden prever atascos). El grafo correspondiente tendrá como vértices las ciudades, como aristas las carreteras y la valuación será la distancia entre ellas. Y, de momento, no se conocen métodos generales para hallar un ciclo de valuación mínima, pero sí para los caminos desde *a* hasta *b*, sin más condición.

Clasificación de Grafos

Como se detallo en el apartado anterior, los grafos pueden clasificarse según diferentes criterios, a continuación describiremos las clasificaciones más relevantes en la administración de datos:

Los grafos en función de si su dirección es definida o no se clasifican en:

- ***Grafos Dirigidos:*** son aquellos en los cuales los arcos que vinculan a los vértices tienen una dirección definida, la cual marca una jerarquía en la relación modelizada.

- ***Grafos No dirigidos:*** son aquellos donde los arcos no tienen una dirección definida que marque propiedad en la relación modelizada.

En función de las restricciones que pueden ser aplicadas a las relaciones que modelizan, los grafos se clasifican en:

Los grafos en función de si su dirección es definida o no se clasifican en:

- ***Grafos Restrictos:*** son aquellos en los cuales la relación modelizada no debe cumplir las propiedades de reflexivididad, simetría y transitividad.

- ***Grafos Irrestrictos:*** son aquellos que pueden modelizar cualquier relación independientemente de las propiedades que cumpla o no.

Representación Computacional de Grafos

Como se observó en los apartados anteriores, los grafos modelizan lógicamente a problemas de la vida real y sirven para graficar las relaciones (arcos) entre los distintos participantes (nodos).

Ahora bien, dichos grafos son abstractos, esto es no existen en la realidad computacional, para existir deben reflejarse a través de una representación computacional. Estas representaciones pueden ser de dos tipos dinámicas y estáticas.

Representaciones Dinámicas

Una representación es dinámica, cuando el espacio consumido para representar computacionalmente al grafo, concuerda exactamente con la cantidad de nodos y vértices a representar, esto es que no se consideran todas las posibilidades de relación posibles, sino que solo se representa lo que ocurre en este momento.

De esta forma la representación sigue la "dinámica" de la estructura y es por ello que las denominamos representaciones dinámicas.

Dinámicamente un grafo irrestricto puede representarse a través de varias estructuras linkeadas, dentro de ellas encontramos a las ***Listas de Adyacencia***, la ***Estructura de Graal*** y la ***Estructura de Pfaltz***.

Listas de adyacencia

Una forma de representar un grafo es por medio de listas que definen las aristas que conectan los nodos. Lo que se hace es definir una lista enlazada para cada nodo, que contendrá los nodos a los cuales es posible acceder. Es decir, un nodo A tendrá una lista enlazada asociada en la que aparecerá un elemento con una referencia al nodo B si A y B tienen una arista que los une. Obviamente, si el grafo es no dirigido, en la lista enlazada de B aparecerá la correspondiente referencia al nodo A.

A continuación, se describe el algoritmo de implementación en lenguaje C de una lista de adyacencia utilizando la estructura nodo.

Algoritmo

```
struct nodo
{
  int v;
  int p;
  nodo *sig;
};

int V,A;  // vértices y aristas del grafo
struct nodo *a[maxV], *z;

void inicializar()
{
  int i,x,y,peso;
  char v1,v2;
  struct nodo *t;
  z=(struct nodo *)malloc(sizeof(struct nodo));
  z->sig=z;
  for (i=0; i<V; i++)
    a[i]=z;
  for (i=0; i<A; i++)
  {
    scanf("%c %c %d\n",&v1,&v2,&peso);
    x=v1-'A'; y=v2-'A';

    t=(struct nodo *)malloc(sizeof(struct nodo));
    t->v=y; t->p=peso; t->sig=a[x]; a[x]=t;

    t=(struct nodo *)malloc(sizeof(struct nodo));
    t->v=x; t->p=peso; t->sig=a[y]; a[y]=t;
  }
}
```

En este caso el espacio ocupado es $O(V + A)$, muy distinto del necesario en la matriz de adyacencia, que era de $O(V^2)$. La representación por listas de adyacencia, por tanto, será más adecuada para grafos dispersos.

Hay que tener en cuenta un aspecto importante y es que la implementación con listas enlazadas determina fuertemente el tratamiento del grafo posterior. Como se puede ver en el código, los nodos se van añadiendo a las listas según se leen las aristas, por lo que nos encontramos que un mismo grafo con un orden distinto de las aristas en la entrada producirá listas de adyacencia diferentes y por ello el orden en que los nodos se procesen variará. Una consecuencia de esto es que si un problema tiene varias soluciones la primera que se encuentre dependerá de la entrada dada. Podría presentarse el caso de tener varias soluciones y tener que mostrarlas siguiendo un determinado orden. Ante una situación así podría ser muy conveniente modificar la forma de meter los nodos en la lista (por ejemplo, hacerlo al final y no al principio, o incluso insertarlo en una posición adecuada), de manera que el algoritmo mismo diera las soluciones ya ordenadas.

Representaciones Estáticas

Una representación computacional se denomina estática, cuando el espacio consumido para representar computacionalmente al grafo es invariable y fijo respecto a la cantidad de nodos y vértices a representar, esto es que son consideradas todas las ocurrencias de relaciones que puedan producirse entre todos los nodos, reservando el espacio para dicha ocurrencia potencial.

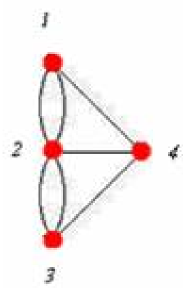
De esta forma la representación no sigue la “dinámica” de la estructura, sino que se mantiene fija y estática en el tiempo independientemente de las altas y bajas de arcos o vértices que se produzcan en el grafo representado computacionalmente.

Estáticamente un grafo irrestricto puede representarse a través de:

- **Matriz de adyacencia**
- **Matriz de incidencia**

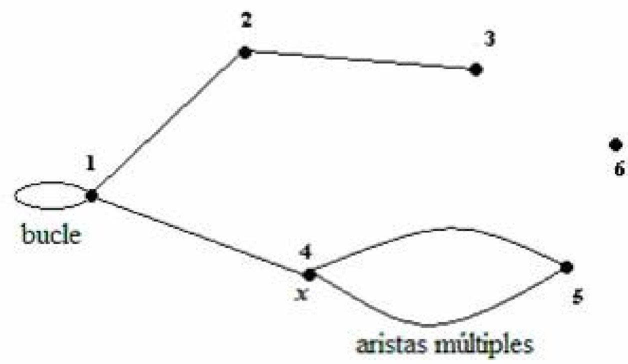
Matriz de adyacencia

Dado un grafo G = (V, A) con n vértices $\{V_1, ..., V_n\}$ su **matriz de adyacencia** es la matriz de orden nxn, $A(G)=(A_{ij})$ donde A_{ij} es el número de aristas que unen los vértices V_i y V_j.



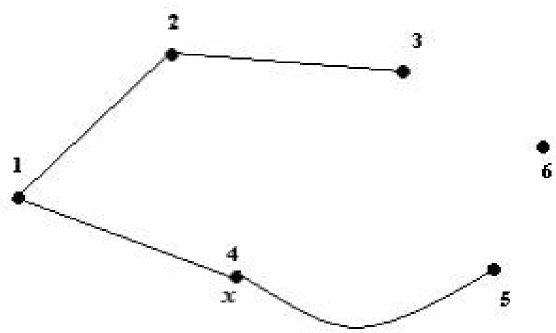
$$A(G) = \begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 \\ 0 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

G



$$A(H) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

H

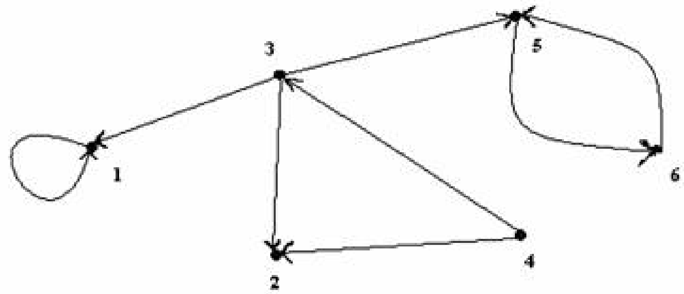


$$A(H) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

H'

La matriz de adyacencia de un grafo es simétrica. Si un vértice es aislado entonces la correspondiente fila (columna) esta compuesta sólo por ceros. Si el grafo es simple entonces la matriz de adyacencia contiene solo ceros y unos (matriz binaria) y la diagonal esta compuesta sólo por ceros.

Dado un grafo dirigido o dígrafo D = (V, A) con n vértices $\{V_1, ..., V_n\}$ su **matriz de adyacencia** es la matriz de orden nxn, $A(D)=(A_{ij})$ donde A_{ij} es el número de arcos que tienen a V_i como extremo inicial y a V_j como extremo final.

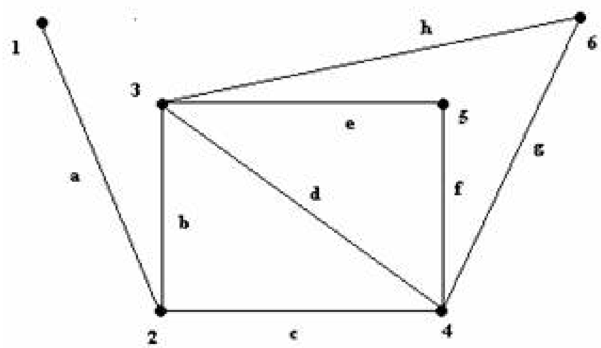


$$A(D) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriz de adyacencia de un dígrafo no es simétrica. Es una matriz binaria. El número de unos que aparecen en una fila es igual al grado de salida del correspondiente vértice y el número de unos que aparecen en una determinada columna es igual al grado de entrada del correspondiente vértice.

Matriz de incidencia

Dado un grafo simple G = (V, A) con n vértices $V = \{V_1, ..., V_n\}$ y m = aristas $A = \{A_1, ..., A_m\}$, su **matriz de incidencia** es la matriz de orden nxm, $B(G)=(b_{ij})$, donde $b_{ij}=1$ si V_i es incidente con A_j y bij=0 en caso contrario.



$$B = \begin{array}{c} \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}$$

La matriz de incidencia sólo contiene ceros y unos (matriz binaria). Como cada arista incide exactamente en dos vértices, cada columna tiene exactamente dos unos. El número de unos que aparece en cada fila es igual al grado del vértice correspondiente. Una fila compuesta sólo por ceros corresponde a un vértice aislado.