Teoria dos Grafos

Valeriano A. de Oliveira
Socorro Rangel
Departamento de Matemática Aplicada

antunes@ibilce.unesp.br, socorro@ibilce.unesp.br

Representação de Grafos

Preparado a partir do texto:
Rangel, Socorro. Teoria do Grafos, Notas de aula, IBILCE, Unesp, 2002-2013.

Representação de Grafos

A representação computacional de um grafo (ou digrafo) deve usar uma estrutura que:

- corresponde de forma única a um grafo dado;
- pode ser armazenada e manipulada em um computador.

A representação gráfica de um grafo através do diagrama de pontos e linhas não satisfaz a segunda condição acima.

Vamos discutir a seguir algumas estruturas que satisfazem estes dois critérios.

Considere um grafo $G(V, A)$ com n vértices e m arestas.

Matriz de Adjacência

É uma matriz $n \times n$, denotada por $X = [x_{ij}]$ e definida como:

$$x_{ij} = \begin{cases} 1 & \text{se existe uma aresta entre os vértices } v_i \text{ e } v_j, \\ 0 & \text{caso contrário.} \end{cases}$$

Observações:

- É necessário fazer uma rotulação nos vértices de G.
- A complexidade em termos de espaço de memória de um algoritmo que use uma matriz de adjacência para armazenar o grafo é $O(n^2)$.
- Qualquer tipo de grafo pode ser armazenado nesta estrutura?

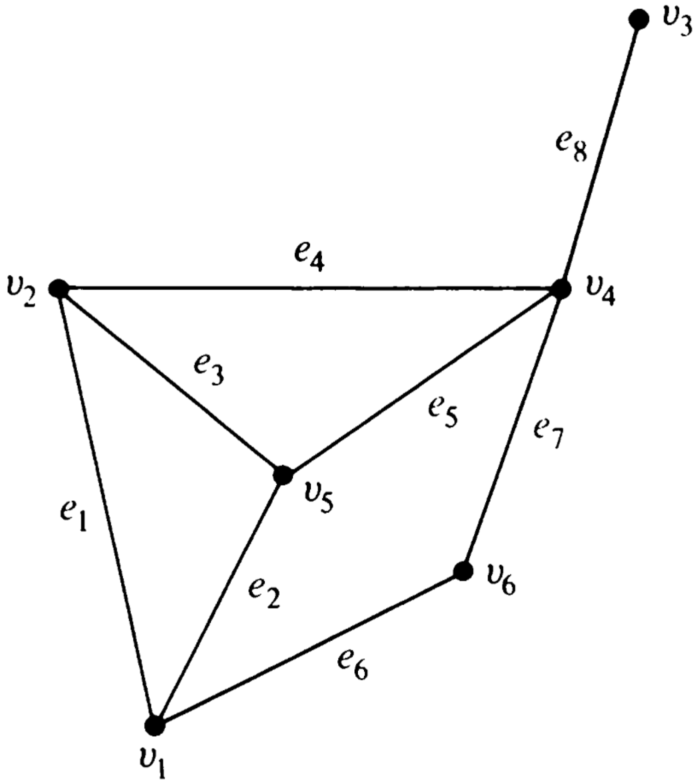
Matriz de Adjacência

É uma matriz $n \times n$, denotada por $X = [x_{ij}]$ e definida como:

$$x_{ij} = \begin{cases} 1 & \text{se existe uma aresta entre os vértices } v_i \text{ e } v_j, \\ 0 & \text{caso contrário.} \end{cases}$$

Observações:

- É necessário fazer uma rotulação nos vértices de G.
- A complexidade em termos de espaço de memória de um algoritmo que use uma matriz de adjacência para armazenar o grafo é $O(n^2)$.
- Qualquer tipo de grafo pode ser armazenado nesta estrutura? Não! Apenas grafos que não possuam arestas paralelas.



$$X = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Observações

- As entradas ao longo da diagonal principal de X são todas nulas se, e somente se, o grafo não possui laços. Quando há um laço em um vértice v_i temos $x_{ii} = 1$.

Observações

- As entradas ao longo da diagonal principal de X são todas nulas se, e somente se, o grafo não possui laços. Quando há um laço em um vértice v_i temos $x_{ii} = 1$.
- Se o grafo é simples, o grau de um vértice é dado pela soma dos elementos de sua linha (ou coluna) correspondente.

Observações

- As entradas ao longo da diagonal principal de X são todas nulas se, e somente se, o grafo não possui laços. Quando há um laço em um vértice v_i temos $x_{ii} = 1$.
- Se o grafo é simples, o grau de um vértice é dado pela soma dos elementos de sua linha (ou coluna) correspondente.
- Permutações de linhas e das colunas correspondentes implicam em uma reordenação dos vértices. Portanto dois grafos simples G_1 e G_2 são isomorfos se, e somente se, $X(G_2) = R^{-1}X(G_1)R$, onde R é uma matriz de permutação.

Observações

- As entradas ao longo da diagonal principal de X são todas nulas se, e somente se, o grafo não possui laços. Quando há um laço em um vértice v_i temos $x_{ii} = 1$.
- Se o grafo é simples, o grau de um vértice é dado pela soma dos elementos de sua linha (ou coluna) correspondente.
- Permutações de linhas e das colunas correspondentes implicam em uma reordenação dos vértices. Portanto dois grafos simples G_1 e G_2 são isomorfos se, e somente se, $X(G_2) = R^{-1}X(G_1)R$, onde R é uma matriz de permutação.
- Dado uma matriz qualquer $Q \in \mathbb{R}^{n\times n}$ simétrica e binária, sempre é possível construir um grafo G com n vértices tal que $X(G) = Q$.

Observações

- As entradas ao longo da diagonal principal de X são todas nulas se, e somente se, o grafo não possui laços. Quando há um laço em um vértice v_i temos $x_{ii} = 1$.
- Se o grafo é simples, o grau de um vértice é dado pela soma dos elementos de sua linha (ou coluna) correspondente.
- Permutações de linhas e das colunas correspondentes implicam em uma reordenação dos vértices. Portanto dois grafos simples G_1 e G_2 são isomorfos se, e somente se, $X(G_2) = R^{-1}X(G_1)R$, onde R é uma matriz de permutação.
- Dado uma matriz qualquer $Q \in \mathbb{R}^{n \times n}$ simétrica e binária, sempre é possível construir um grafo G com n vértices tal que $X(G) = Q$.
- Quantos elementos diferentes de zero esta matriz possui?

- Um grafo G é desconexo com dois componentes G_1 e G_2 se, e somente se,

$$X(G) = \begin{bmatrix} X(G_1) & 0 \\ 0 & X(G_2) \end{bmatrix}.$$

- Um grafo G é desconexo com dois componentes G_1 e G_2 se, e somente se,

$$X(G) = \begin{bmatrix} X(G_1) & 0 \\ 0 & X(G_2) \end{bmatrix}.$$

- O que representa a matriz $B = X^2$? Os elementos $b_{ij},\ i \neq j$, representam o número de caminhos distintos de comprimento 2 entre os vértices v_i e v_j. De fato:

$$b_{ij} = \sum_{k=1}^{n} x_{ik}x_{kj} = x_{i1}x_{1j} + x_{i2}x_{2j} + \ldots + x_{in}x_{nj}$$

e existe um caminho se $x_{ik} = x_{kj} = 1$, e o caminho é dado por $\{i, (i,k), k, (k,j), j\}$.

Teorema 1. *Seja* X *a matriz de adjacência de um grafo simples* G*. Então a* ij*-ésima entrada de* X^r *é o numero de passeios diferentes de comprimento* r *entre os vértices* v_i *e* v_j*.*

Teorema 4. *Seja X a matriz de adjacência de um grafo simples G. Então a ij-ésima entrada de X^r é o numero de passeios diferentes de comprimento r entre os vértices v_i e v_j.*

Corolário 5. *Em um grafo conexo, a distância entre dois vértices v_i e v_j, $i \neq j$, é k se, e somente se, k é o menor inteiro para o qual a ij-ésima entrada em X^k é não-nula.*

Teorema 7. *Seja X a matriz de adjacência de um grafo simples G. Então a ij-ésima entrada de X^r é o numero de passeios diferentes de comprimento r entre os vértices v_i e v_j.*

Corolário 8. *Em um grafo conexo, a distância entre dois vértices v_i e v_j, $i \neq j$, é k se, e somente se, k é o menor inteiro para o qual a ij-ésima entrada em X^k é não-nula.*

Corolário 9. *Se X é a matriz de adjacência de um grafo com n vértices, e*

$$Y = X + X^2 + X^3 + \ldots + X^{n-1},$$

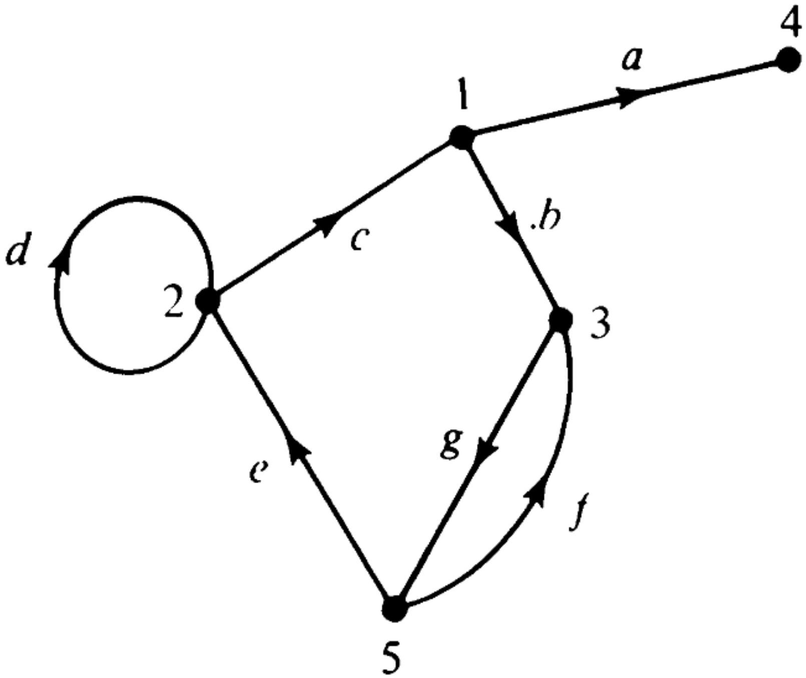
então G é desconexo se, e somente se, existe ao menos uma entrada na matriz Y que é igual a zero.

É possível utilizar esta estrutura para armazenar digrafos?

Sim. Dado um digafo $D(V, A)$ com n vértices e sem arestas paralelas, definimos

$$x_{ij} = \begin{cases} 1 & \text{se existe uma aresta direcionada do vértice } v_i \text{ para o vértice } v_j, \\ 0 & \text{caso contrário.} \end{cases}$$

Exemplo



$$X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Observações

- Neste caso a matriz só será simétrica se o digrafo for simétrico.
- O grau de saída do vértice v_i é dado pela soma dos elementos da linha i.
- O grau de entrada do vértice v_i é dado pela soma dos elementos da coluna i.
- Se X é a matriz de adjacência de um digrafo D, então a sua transposta $X^\top$ é a matriz de adjacência do digrafo obtido pela inversão da orientação das arestas de D.

Matriz de Incidência

Seja G um grafo com n vértices e m arestas. Sua matriz de incidência é uma matriz de ordem $n \times m$, denotada por $A = [a_{ij}]$, definida como

$$x_{ij} = \begin{cases} 1 & \text{se a aresta } a_j \text{ é incidente no } v_i, \\ 0 & \text{caso contrário.} \end{cases}$$

Observações:

- É necessário fazer uma rotulação nos vértices e nas arestas de G.
- A complexidade em termos de espaço de memória de um algoritmo que use uma matriz de adjacência para armazenar o grafo é $O(nm)$.
- Qualquer tipo de grafo pode ser armazenado nesta estrutura?

Matriz de Incidência

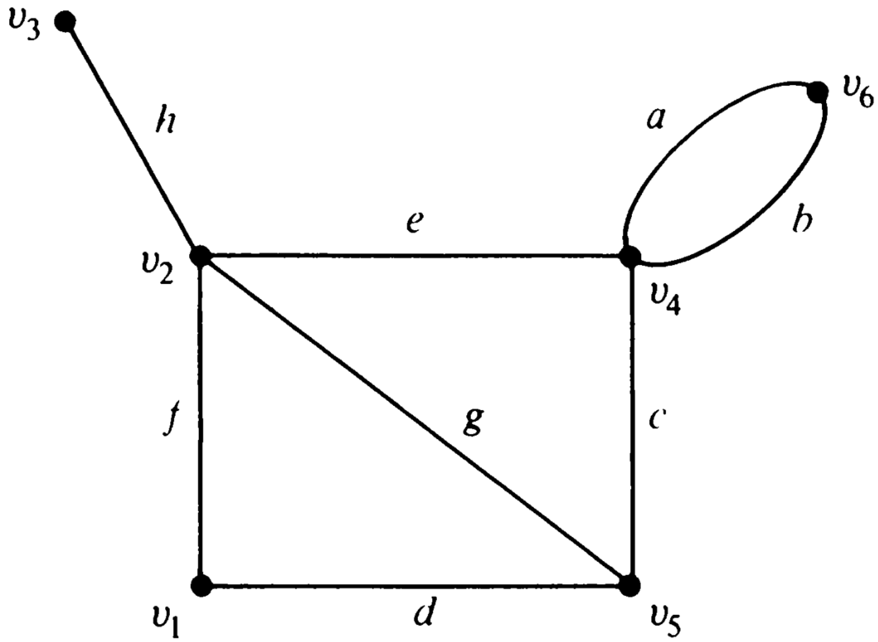
Seja G um grafo com n vértices e m arestas. Sua matriz de incidência é uma matriz de ordem $n \times m$, denotada por $A = [a_{ij}]$, definida como

$$x_{ij} = \begin{cases} 1 & \text{se a aresta } a_j \text{ é incidente no } v_i, \\ 0 & \text{caso contrário.} \end{cases}$$

Observações:

- É necessário fazer uma rotulação nos vértices e nas arestas de G.
- A complexidade em termos de espaço de memória de um algoritmo que use uma matriz de adjacência para armazenar o grafo é $O(nm)$.
- Qualquer tipo de grafo pode ser armazenado nesta estrutura? Não! Apenas grafos que não possuam arestas laço.

Exemplo



$$
\begin{array}{c}
\begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

Observações

- Como cada aresta é incidente em exatamente dois vértices, cada coluna de $A(G)$ possui exatamente dois 1's.
- O número de 1's em cada linha é igual ao grau do vértice correspondente.
- Uma linha de 0's representa um vértice isolado.
- Arestas paralelas correspondem a colunas idênticas.
- Um grafo G é desconexo com dois componentes G_1 e G_2, então

$$A(G) = \begin{bmatrix} A(G_1) & 0 \\ 0 & A(G_2) \end{bmatrix}.$$

- Dois grafos G_1 e G_2 são isomorfos se, e somente se, suas matrizes de incidência $A(G_1)$ e $A(G_2)$ diferem apenas por permutações de linhas e colunas.
- Quantos elementos diferentes de zero esta matriz possui?

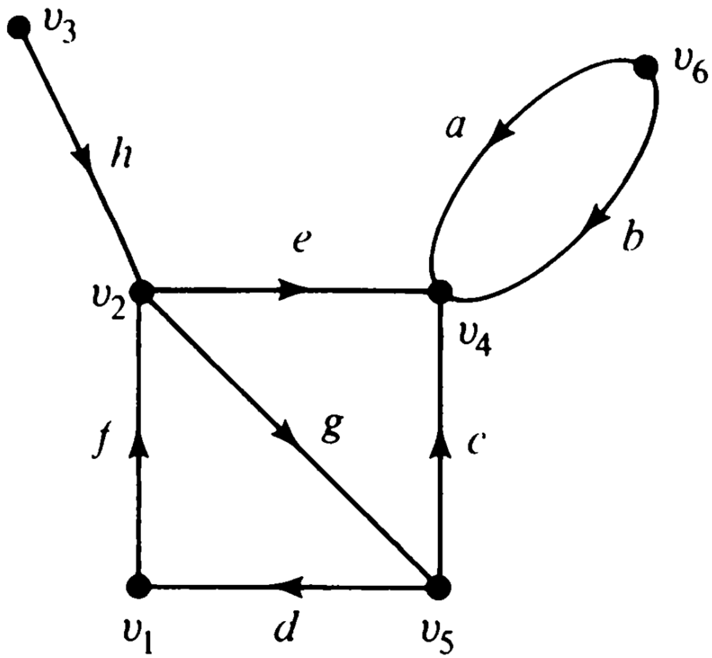
É possível utilizar esta estrutura para armazenar digrafos?

Sim. Com uma pequena modificação, uma vez que ao dizer que uma aresta incide em um vértice é necessário especificar se ela converge para ou diverge para este vértice.

Seja D um digrafo com n vértices e m arestas e sem arestas laço. Sua matriz de incidência $A = [a_{ij}] \in \mathbb{R}^{n \times m}$ é definida como

$$x_{ij} = \begin{cases} 1 & \text{se a aresta } a_j \text{ diverge do vértice } v_i, \\ -1 & \text{se a aresta } a_j \text{ converge para o vértice } v_i, \\ 0 & \text{caso contrário.} \end{cases}$$

Exemplo



$$
\begin{array}{c}
 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

As duas representações dadas (matrizes de adjacência e de incidência) são importantes porque elas facilitam a recuperação de uma série de informações a respeito de um grafo.

Por exemplo o grau de um vértice, determinar se dois vértices são adjacentes, entre outras.

No entanto elas não valem para qualquer grafo dado, e demandam muito espaço de memória: $O(n^2)$ e $O(nm)$ para armazenar apenas $2m$ elementos diferentes de zero.

É possível encontrar formar mais eficientes de armazenamento de dados.

É bom salientar no entanto que a melhor maneira de armazenar um grafo ou digrafo vai depender do algoritmo a ser implementado.

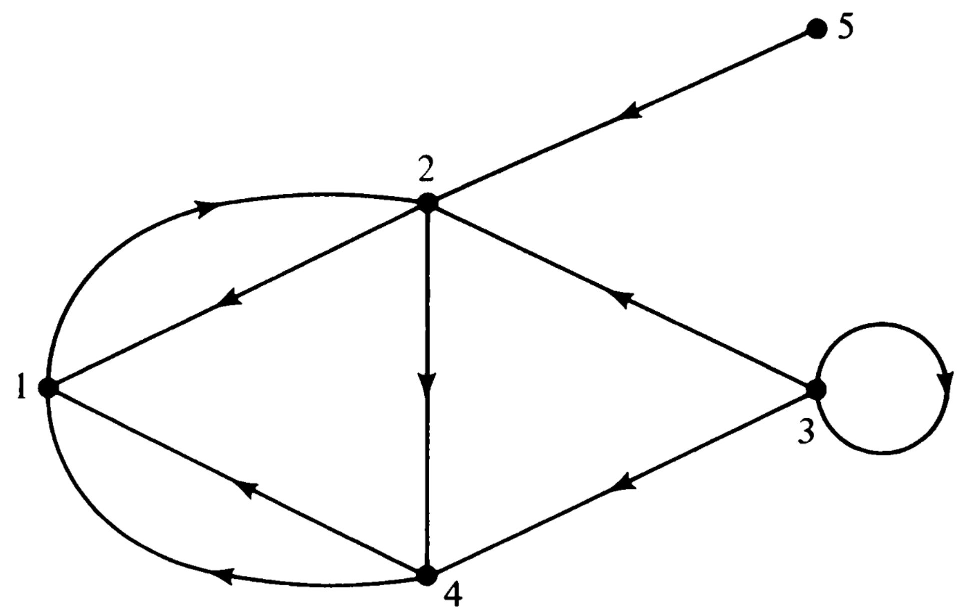
Lista de Arestas

O digrafo (ou grafo) G é representado por dois vetores m-dimesionais $F = (f_1, f_2, \ldots, f_m)$ e $H = (h_1, h_2, \ldots, h_m)$.

Cada elemento destes vetores recebe o rótulo de um vértice, de modo que a i-ésima aresta diverge do vértice f_i e converge para o vértice h_i.

Qual é o espaço necessário para esta estrutura? $O(2m)$.

Exemplo



$$F = (5, 2, 1, 3, 2, 4, 4, 3, 3),$$

$$H = (2, 1, 2, 2, 4, 1, 1, 4, 3).$$

Lista de sucessores

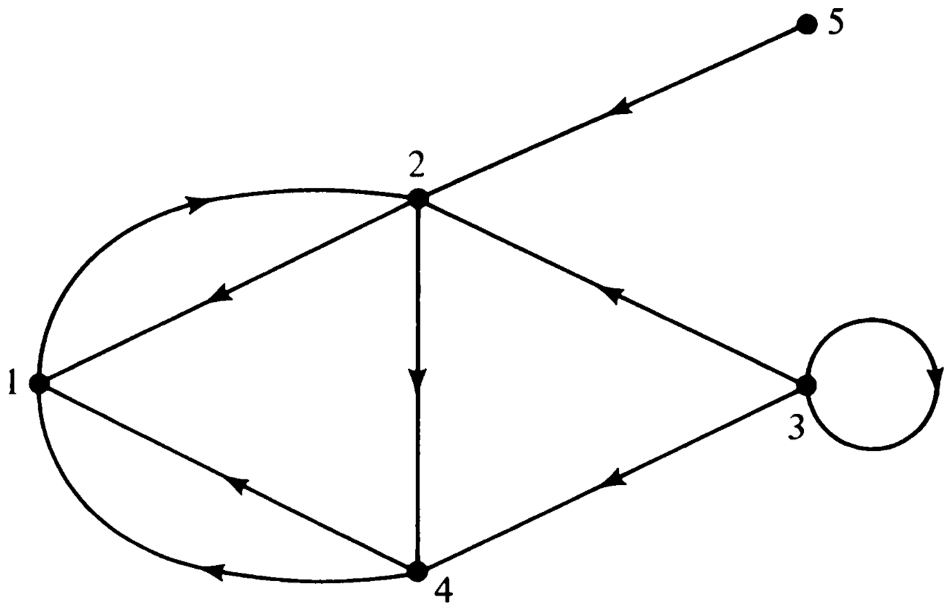
Quando a razão m/n não é muito alta, é conveniente usar uma lista de sucessores.

Para isto definimos n vetores. Cada vetor é associado a um vértice.

O primeiro elemento do vetor k é o vértice v_k e os demais elementos são os vértices adjacentes ao vértice v_k (em um digrafo, os vértice que estão ligados ao vértice k por um caminho de comprimento 1).

Supondo que d_{med} é o grau médio (ou grau de saída médio), o espaço de memória necessário para esta estrutura é $O(nd_{med})$.

Exemplo



1 : 2

2 : 1, 4

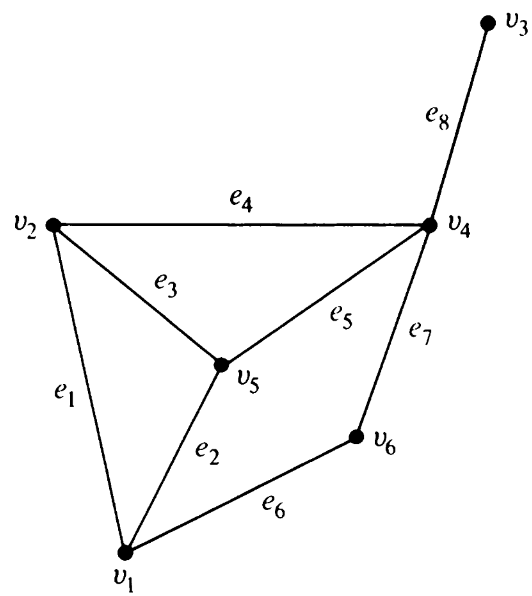
3 : 2, 3, 4

4 : 1, 1

5 : 2

Exercício

- Construa a lista de arestas e a lista de sucessores para o grafo abaixo.



- Escreva um algoritmo para transformar a matriz de incidência de um grafo (ou digrafo) na matriz de adjacências. Codifique o algoritmo na sua linguagem de programação favorita. Teste com grafos de diversas densidades. Qual é a complexidade computacional do seu algoritmo?