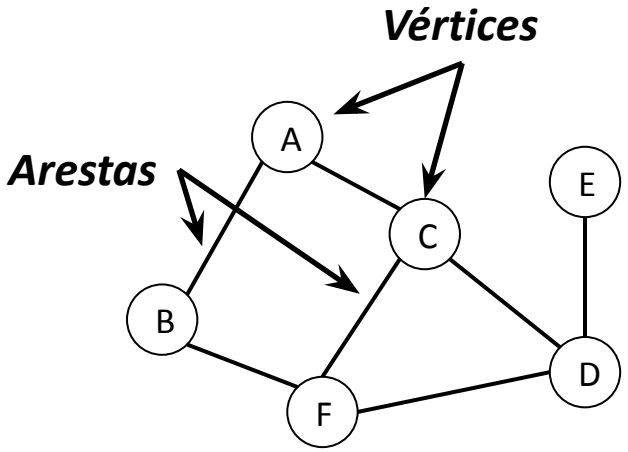
Análise de grafos parte 2:
matriz de incidência e lista de adjacência

Retomando...

Sistemas para Internet
UFSM

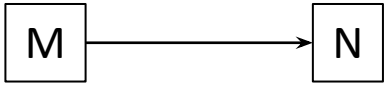
- Grafos:
  - Formados por nós (vértices) e arcos (arestas):
    - G = (V, A);
  - Sequência de nós. G = {A, B, C, D, E, F};
  - Grafo não-orientado: G = {(A, B), (A, C) ...};
  - Grafo orientado: G = {<A, B>, <A, C> ...};
  - Incidência: nós que compõem um arco;
  - Grau: número de arcos que são incididos por um nó.

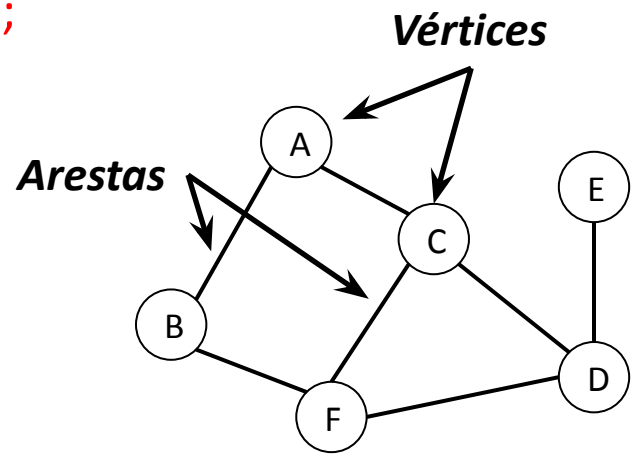


Retomando...

Sistemas para Internet
UFSM

- Grafos:
  - **Adjacente**: nó que recebe o destino de um arco;



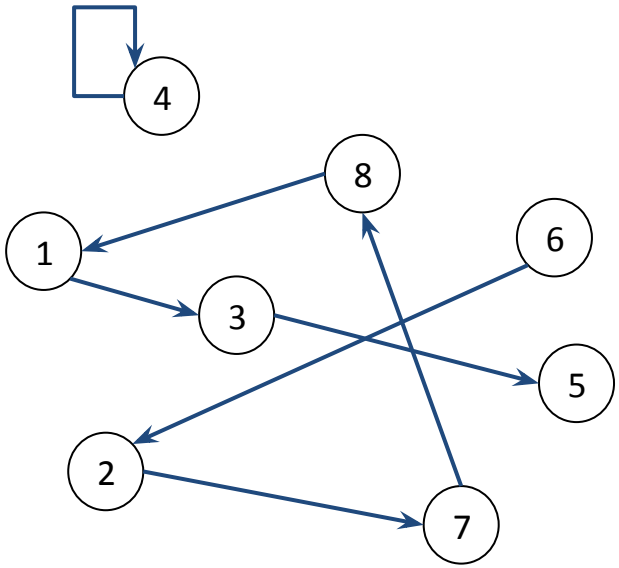


  - **Relação** é a sequência de pares do grafo;
  - Grafo **ponderado**: arcos podem ter pesos;
  - **Caminho**: distância entre dois nós quaisquer;
  - **Comprimento:** Número de arestas que pertencem a um caminho;
  - **Ciclo** é o caminho de um nó para ele mesmo.

Retomando...

Sistemas para Internet
UFSM

- Matriz de adjacência:
- G = {1, 2, 3, 4, 5, 6, 7, 8}
- G = {<1, 3>, <3, 5>, <6, 2>, <2, 7>, <7, 8>, <8, 1>, <4, 4>}



destino

origem	1	2	3	4	5	6	7	8
1	0	0	**1**	0	0	0	0	0
2	0	0	0	0	0	0	**1**	0
3	0	0	0	0	**1**	0	0	0
4	0	0	0	**1**	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	**1**	0	0	0	0	0	0
7	0	0	0	0	0	0	0	**1**
8	**1**	0	0	0	0	0	0	0

Matriz de incidência

Sistemas para Internet
UFSM

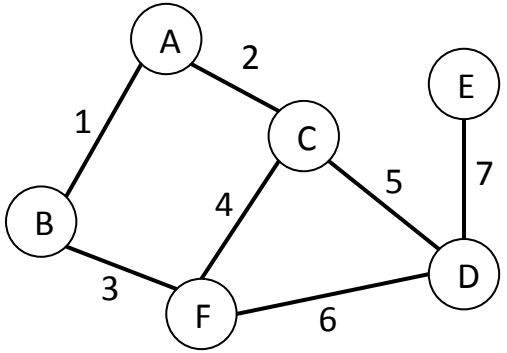
- Definida associando linhas para os nós (N) e colunas para os arcos (M);
- Cada elemento da matriz indica se o arco incide sobre um nó;
- Dada uma matriz A (N x M) de grafo não-orientado:
  - A_{ij} = 1, se o nó ***i*** incide sobre o arco ***j***;
  - A_{ij} = 0, se o nó ***i*** não incide sobre o arco ***j***.
- Dada uma matriz A (N x M) de grafo orientado:
  - A_{ij} = 1, se o nó ***i*** é o nó predecessor do arco ***j***;
  - A_{ij} = 0, se o nó ***i*** não incide sobre o arco ***j***;
  - A_{ij} = -1, se o nó ***i*** é o nó sucessor do arco ***j***.

Matriz de incidência para grafo não-orientado

Sistemas para Internet
UFSM

- Matriz N x M: qual a dimensão da matriz?
  - 6 x 7

		arco						
		1	2	3	4	5	6	7
nó	A	**1**	**1**	0	0	0	0	0
	B	**1**	0	**1**	0	0	0	0
	C	0	**1**	0	**1**	**1**	0	0
	D	0	0	0	0	**1**	**1**	**1**
	E	0	0	0	0	0	0	**1**
	F	0	0	**1**	**1**	0	**1**	0

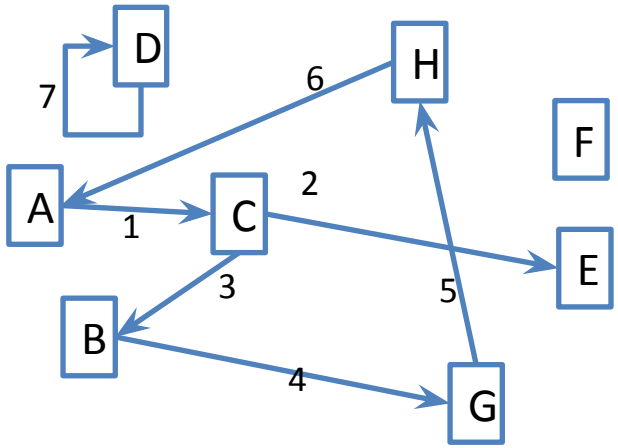


Neste caso os valores dos arcos não referem-se a pesos. São usados para identificá-los.

Matriz de incidência para grafo orientado

Sistemas para Internet
UFSM

- matriz 8 x 7:



		arco						
		1	2	3	4	5	6	7
nó	A	1	0	0	0	0	-1	0
	B	0	0	-1	1	0	0	0
	C	-1	1	1	0	0	0	0
	D	0	0	0	0	0	0	1
	E	0	-1	0	0	0	0	0
	F	0	0	0	0	0	0	0
	G	0	0	0	-1	1	0	0
	H	0	0	0	0	-1	1	0

Desvantagens da representação por matrizes

- Grafos podem ser **esparsos**: muitos vértices e poucas arestas.
  - Consumo desnecessário de memória pois a maioria dos elementos da matriz será 0.

- Matriz de adjacência deve ser quadrática:
  - Exemplo: N x N, onde N é o total de vértices.

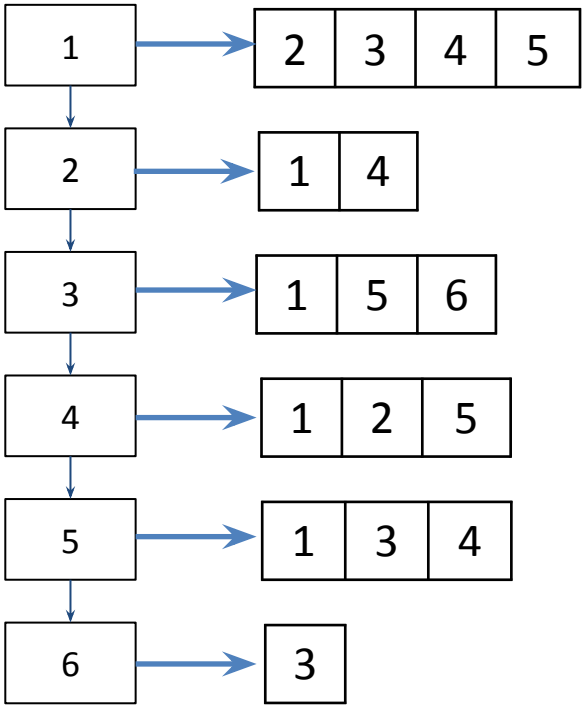
- Matriz de incidência: deve ter a dimensão de Vértices X Arcos;

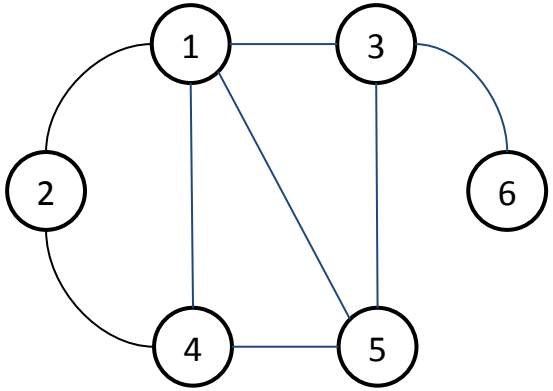
- Alternativa?
  - Representação com listas encadeadas.

Lista de adjacência: grafo não orientado

Sistemas para Internet
UFSM

- Considera-se os todos os pares de cada nó;
- Cria-se uma lista (ou vetor) referente ao total de nós;
- Cada elemento da lista refere-se a um ponteiro para suas conexões (representados com vetor ou lista);
- Total de conexões deve ser o GRAU daquele nó.

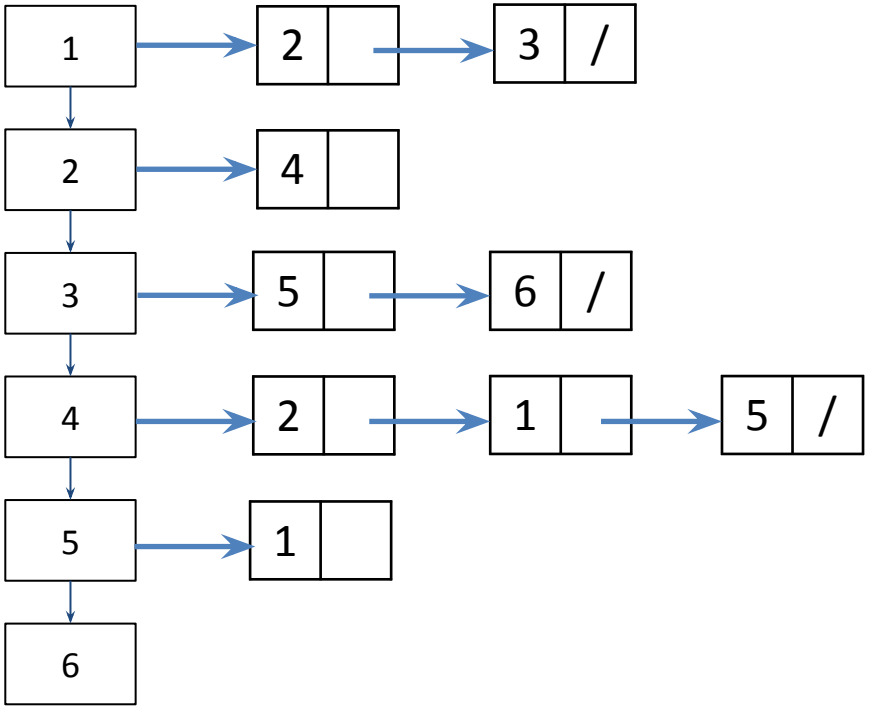


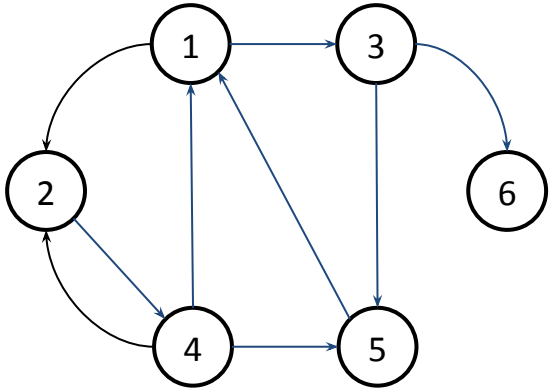


Lista de adjacência: grafo orientado

Sistemas para Internet
UFSM

- Deve-se diferenciar incidência de origem e destino (adjacência);
- Cria-se uma lista (ou vetor) referente ao total de nós;
- Cada elemento (E) da lista refere-se a um ponteiro para uma de suas incidências (I) onde ele é origem. Este (I), por sua vez, aponta para outra conexão de E. O processo se repete para os demais;
- Total de conexões de cada nó corresponde a quantidade de incidências no qual ele é origem.



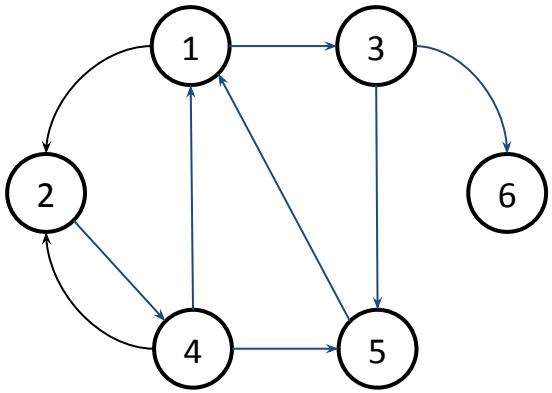


Representação em C

Sistemas para Internet
UFSM

- Criar um registro com os elementos essenciais do grafo;
- Criar um campo do tipo vetor, onde a dimensão é igual ao nó de maior grau.

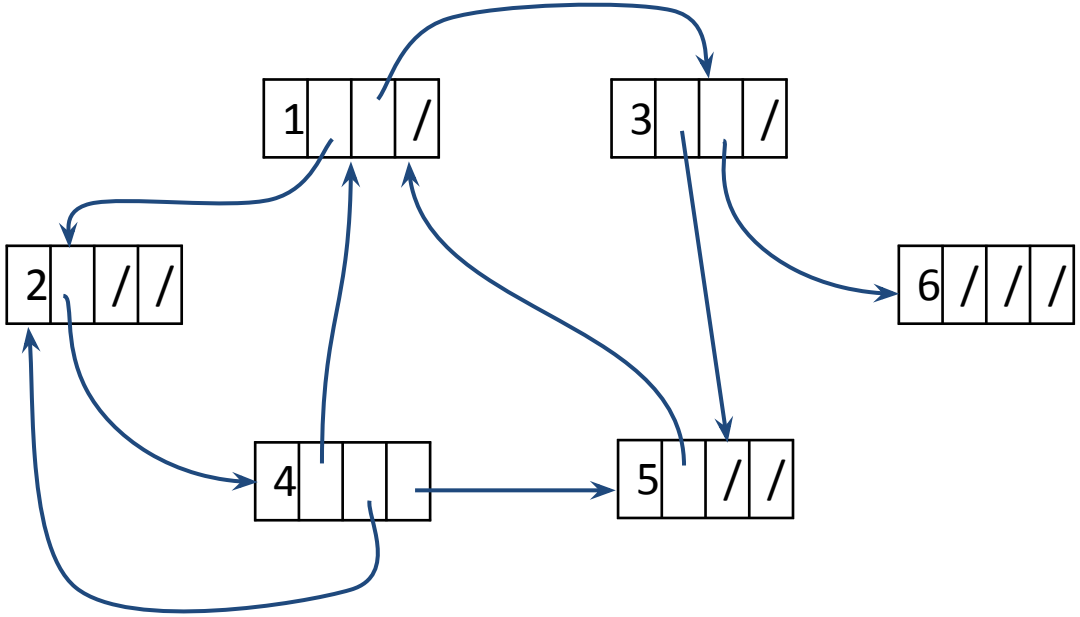
```
struct grafo {
    int no;
    int peso[3]; //se as arestas possuírem peso
    struct grafo *aresta[3];
};
typedef struct grafo g;
```

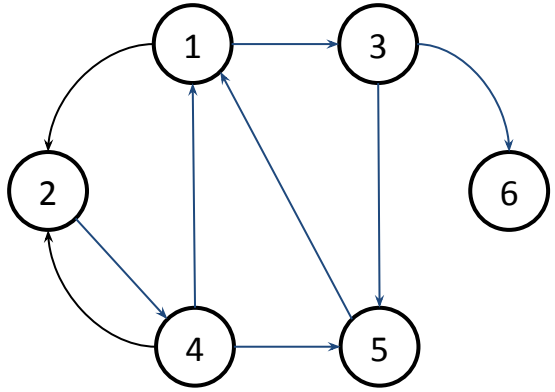


Resultado da aplicação

Sistemas para Internet
UFSM

- Os elementos do vetor são ponteiros para seus vizinhos






```
struct grafo {
    int no;
    // int peso[3];
    struct grafo *aresta[3];
};
typedef struct grafo g;
```

Não utilizado pois as arestas não possuem peso.

Exercício

Sistemas para Internet
UFSM

- Utilizando os conceitos estudados, desenvolva um algoritmo para implementar um grafo com 6 vértices, onde pelo menos dois deles devem ter graus 2 e 3.