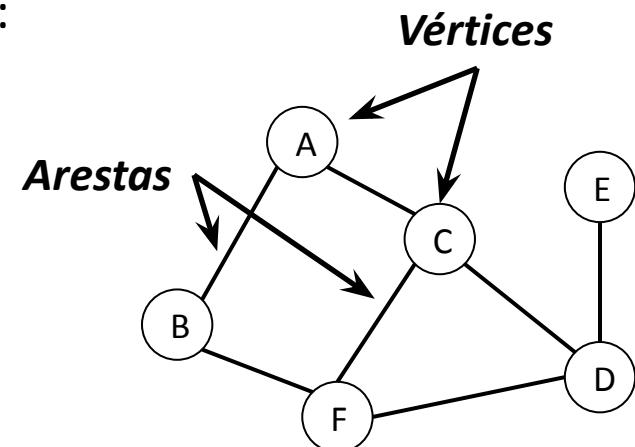


Análise de grafos parte 2: matriz de incidência e lista de adjacência

Retomando...

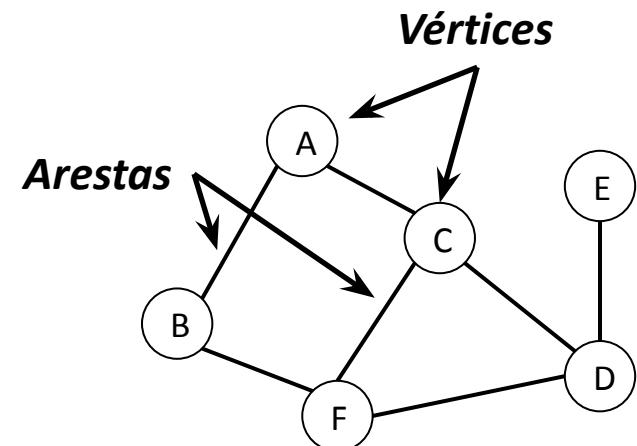
- Grafos:
 - Formados por nós (vértices) e arcos (arestas):
 - $G = (V, A)$;
 - Sequência de nós. $G = \{A, B, C, D, E, F\}$;
 - Grafo não-orientado: $G = \{(A, B), (A, C) \dots\}$;
 - Grafo orientado: $G = \{<A, B>, <A, C> \dots\}$;
 - Incidência: arcos que ligam dois nós:
 - Exemplo: par ordenado (A, C) é incidente de A e incidente a C .
 - Grau: número de arcos que são incididos por um nó.



Retomando...

- Grafos:

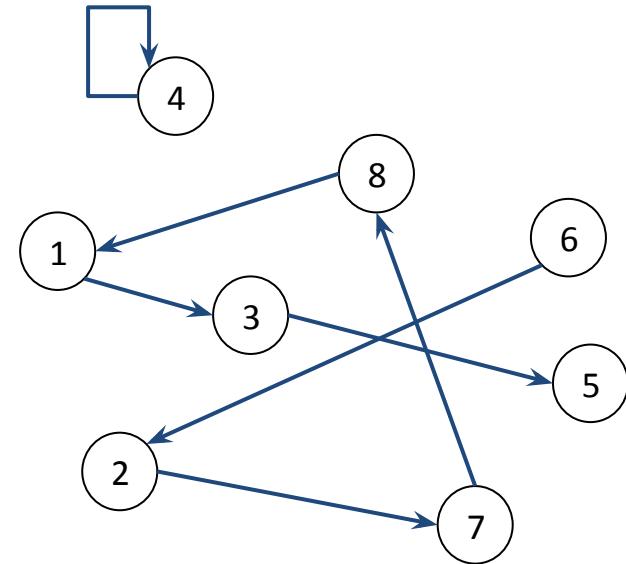
- **Adjacência**: nós conectados por um arco:
 - B, C e D são adjacentes ao nó F.
- **Relação** é a sequência de pares do grafo;
- **Grafo ponderado**: arcos podem ter pesos;
- **Caminho**: distância entre dois nós quaisquer;
- **Comprimento**: Número de arestas que pertencem a um caminho;
- **Ciclo** é o caminho de um nó para ele mesmo.



Retomando...

- Matriz de adjacência:
- $G = \{1, 2, 3, 4, 5, 6, 7, 8\}$
- $G = \{\langle 1, 3 \rangle, \langle 3, 5 \rangle, \langle 6, 2 \rangle, \langle 2, 7 \rangle, \langle 7, 8 \rangle, \langle 8, 1 \rangle, \langle 4, 4 \rangle\}$

		destino							
		1	2	3	4	5	6	7	8
origem	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



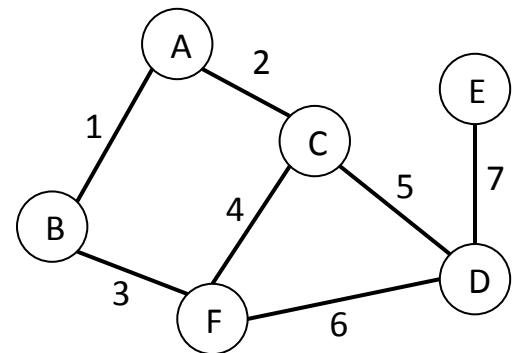
Matriz de incidência

- 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 ($N \times M$) 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 ($N \times M$) 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

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

	arco						
	1	2	3	4	5	6	7
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

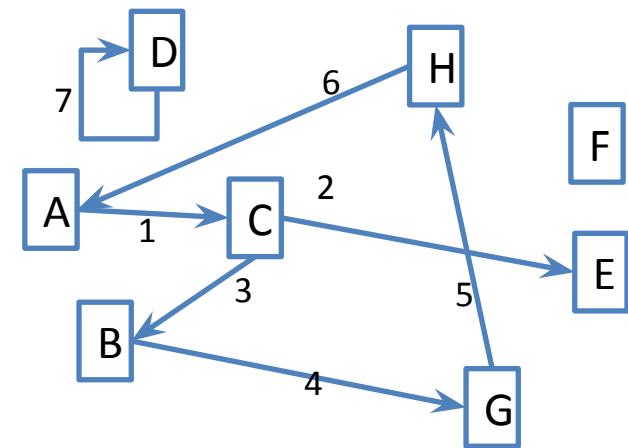


Neste caso os valores dos arcos
não referem-se a pesos.
São usados para identificá-los.

Matriz de incidência para grafo orientado

- matriz 8×7 :

		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

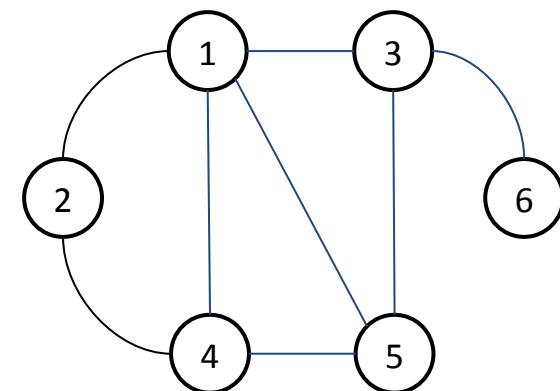
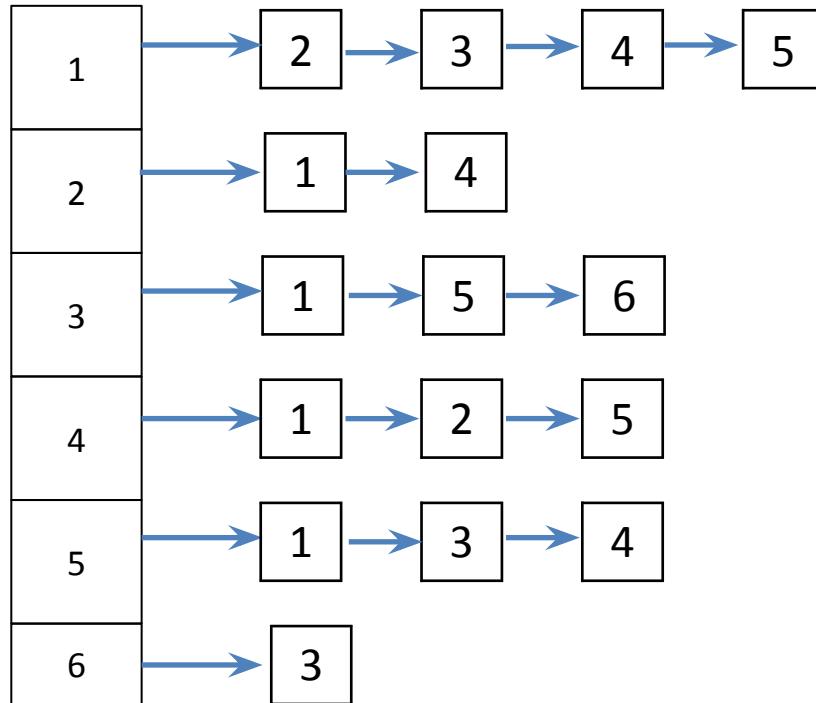


Desvantagens da representação por matrizes

- 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: $N \times N$, onde N é o total de vértices.
- Matriz de incidência: deve ter a dimensão de Vértices X Arcos;
- Alternativa?
 - Representação com listas encadeadas.

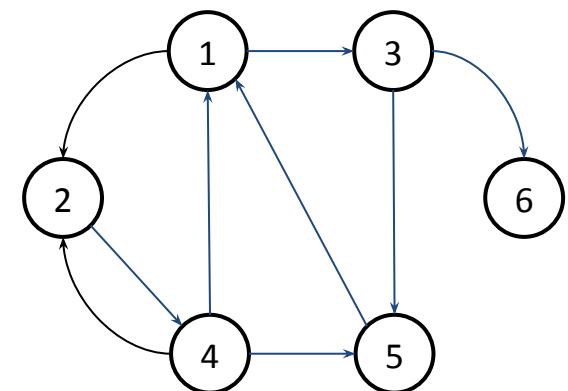
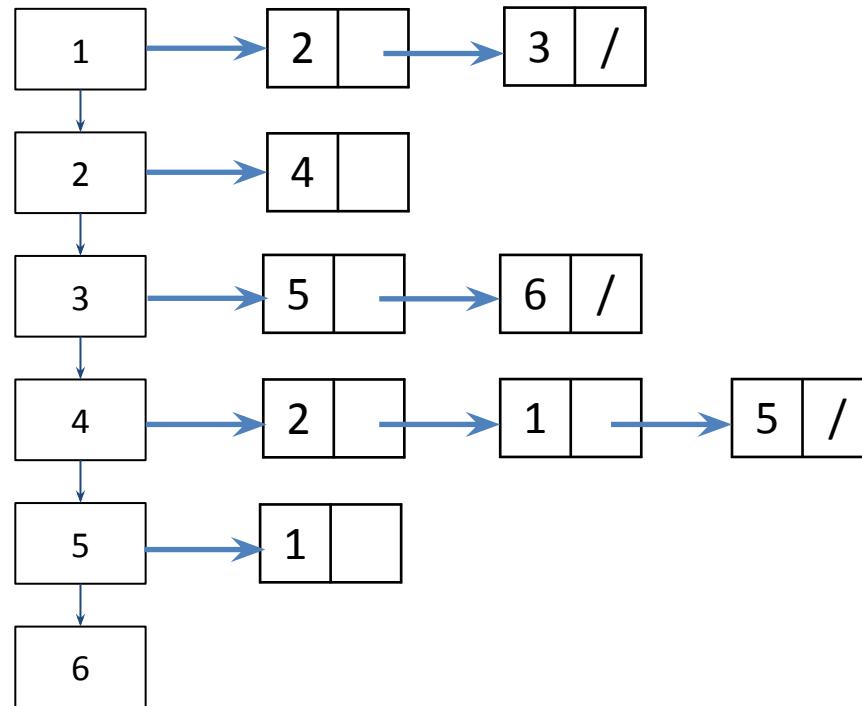
Lista de adjacência: grafo não orientado

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



Lista de adjacência: grafo orientado

- 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.

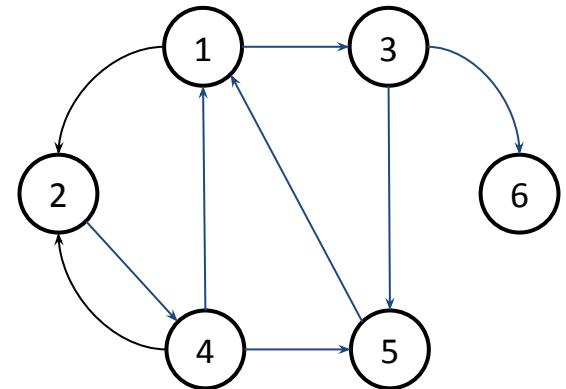


Representação em C

- Com dois registros:
 - Um para o conjunto de nós;
 - Um para listas de adjacentes a cada nó.

```
struct NoAdj {
    int id_destino; // 0 ID do vértice vizinho
    int peso;        // Opcional: para grafos ponderados
    struct NoAdj* prox;
};
```

```
struct Grafo{
    int numVertices;
    NoAdj** adj;
};
```



Criar vetor com o total de vértices.



Exercício

- 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.