

Grafos - Formas de Representação

Prof. Luiz Gustavo Almeida Martins

Grafos: formas de representação

A implementação depende da forma usada para representar o grafo

Questão: como representar um grafo no computador?

Grafos: formas de representação

A implementação depende da forma usada para representar o grafo

Questão: como representar um grafo no computador?

Através da representação de suas **arestas**

Grafos: formas de representação

A implementação depende da forma usada para representar o grafo

Questão: como representar um grafo no computador?

Através da representação de suas **arestas**

As formas mais utilizadas baseiam-se na **relação de adjacência** entre os vértices

Grafos: formas de representação

A implementação depende da forma usada para representar o grafo

Questão: como representar um grafo no computador?

Através da representação de suas **arestas**

As formas mais utilizadas baseiam-se na **relação de adjacência** entre os vértices

Estática: **matriz de adjacências**

Grafos: formas de representação

A implementação depende da forma usada para representar o grafo

Questão: como representar um grafo no computador?

Através da representação de suas **arestas**

As formas mais utilizadas baseiam-se na **relação de adjacência** entre os vértices

Estática: **matriz de adjacências**

Dinâmica: **listas de adjacências**

Grafos: matriz de adjacência

Um **grafo não ponderado** com N vértices é representado através de uma **matriz binária $N \times N$**

Grafos: matriz de adjacência

Um **grafo não ponderado** com N vértices é representado através de uma **matriz binária $N \times N$**

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se V_i e V_j estão ligados por uma aresta

$a_{ij} = 0$, caso contrário

Grafos: matriz de adjacência

Um **grafo não ponderado** com N vértices é representado através de uma **matriz binária** $N \times N$

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se V_i e V_j estão ligados por uma aresta

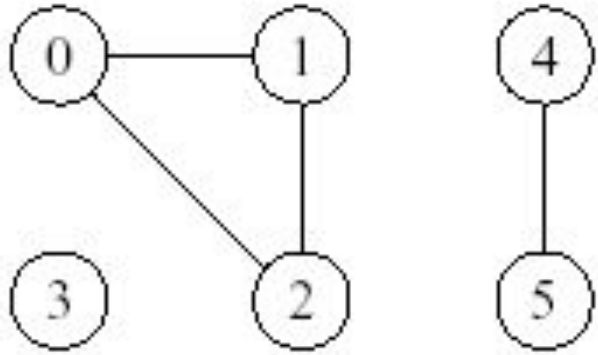
$a_{ij} = 0$, caso contrário

No **grafo ponderado**, o elemento a_{ij} recebe o peso associado com a aresta entre os vértices V_i e V_j

Matriz não é de *bits*

Deve-se adotar um **valor diferenciado** para indicar a inexistência de arestas

Exemplo de matrizes de adjacência

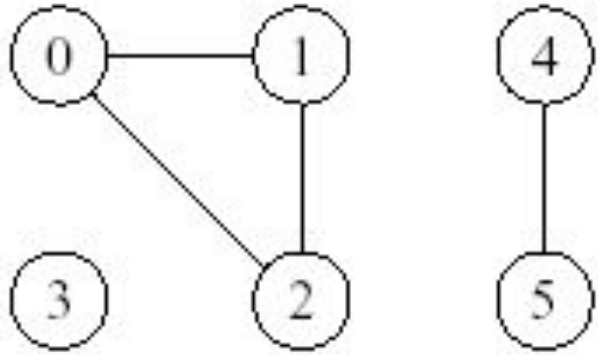


**Grafo não direcionado
e não ponderado**

Matriz de adjacência

	0	1	2	3	4	5
0	0	1	1	0	0	0
1	1	0	1	0	0	0
2	1	1	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	1
5	0	0	0	0	1	0

Exemplo de matrizes de adjacência

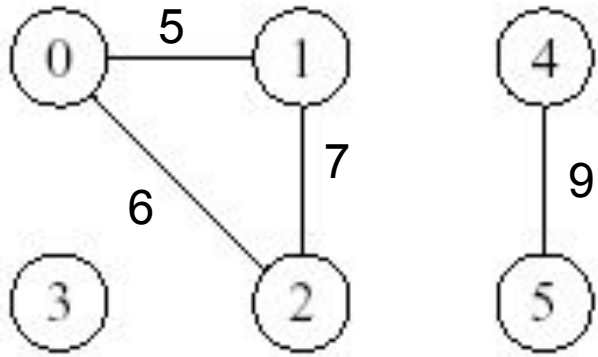


Grafo não direcionado

**Matriz de adjacência
(simétrica)**

	0	1	2	3	4	5
0	0	1	1	0	0	0
1	1	0	1	0	0	0
2	1	1	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	1
5	0	0	0	0	1	0

Exemplo de matrizes de adjacência

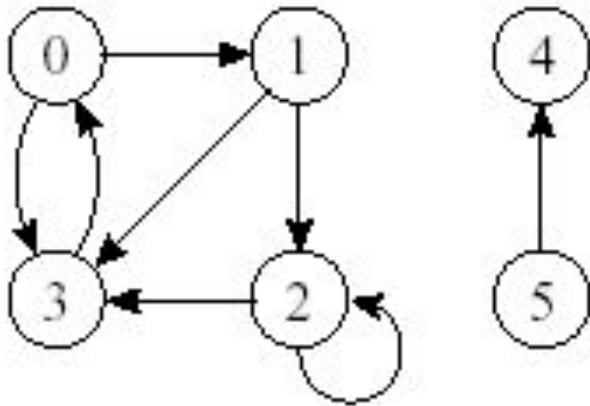


Grafo não direcionado
ponderado

Matriz de adjacência
(simétrica)

	0	1	2	3	4	5
0	0	5	6	0	0	0
1	5	0	7	0	0	0
2	6	7	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	9
5	0	0	0	0	9	0

Exemplo de matrizes de adjacência

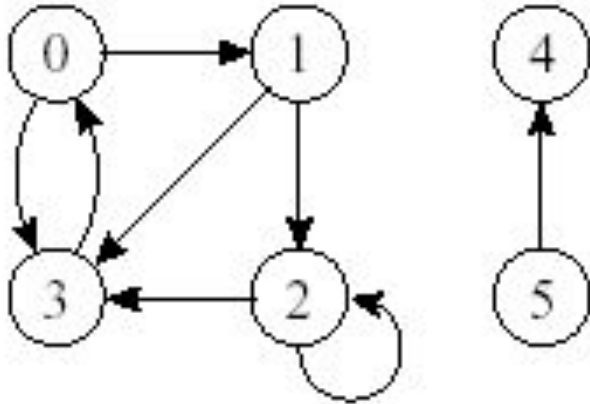


Grafo direcionado

Matriz de adjacência

	0	1	2	3	4	5
0	0	1	0	1	0	0
1	0	0	1	1	0	0
2	0	0	1	1	0	0
3	1	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	1	0

Exemplo de matrizes de adjacência

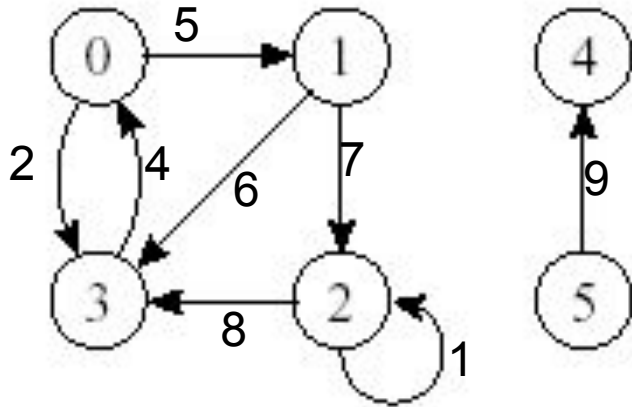


Grafo direcionado

Matriz de adjacência
(não simétrica)

	0	1	2	3	4	5
0	0	1	0	1	0	0
1	0	0	1	1	0	0
2	0	0	1	1	0	0
3	1	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	1	0

Exemplo de matrizes de adjacência



Grafo direcionado
ponderado

Matriz de adjacência
(não simétrica)

	0	1	2	3	4	5
0	0	5	0	2	0	0
1	0	0	7	6	0	0
2	0	0	1	8	0	0
3	4	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	9	0

Grafos: matriz de adjacência

Redundância na representação de grafos não direcionados (**matriz simétrica**)

Grafos: matriz de adjacência

Redundância na representação de grafos não direcionados (**matriz simétrica**)

Possui um **alto custo de armazenamento**

Matriz necessita $\Omega(|V|^2)$ de espaço

Deve ser usada em **grafos densos** ($|A| \approx |V|^2$)

Grafos: matriz de adjacência

Redundância na representação de grafos não direcionados (**matriz simétrica**)

Possui um **alto custo de armazenamento**

Matriz necessita $\Omega(|V|^2)$ de espaço

Deve ser usada em **grafos densos**, onde $|A| \approx |V|^2$

Facilita o **acesso a informação**

Tempo de acesso independe de $|V|$ ou $|A|$

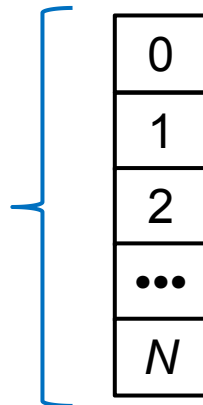
Útil para algoritmos que precisam saber com rapidez se existe uma aresta ligando 2 vértices

Grafos: listas de adjacência

Um **grafo G** com N vértices é representado através de um **vetor de N posições**, onde:

Cada posição refere-se a um vértice V_i

Grafo G
(esquematicamente)

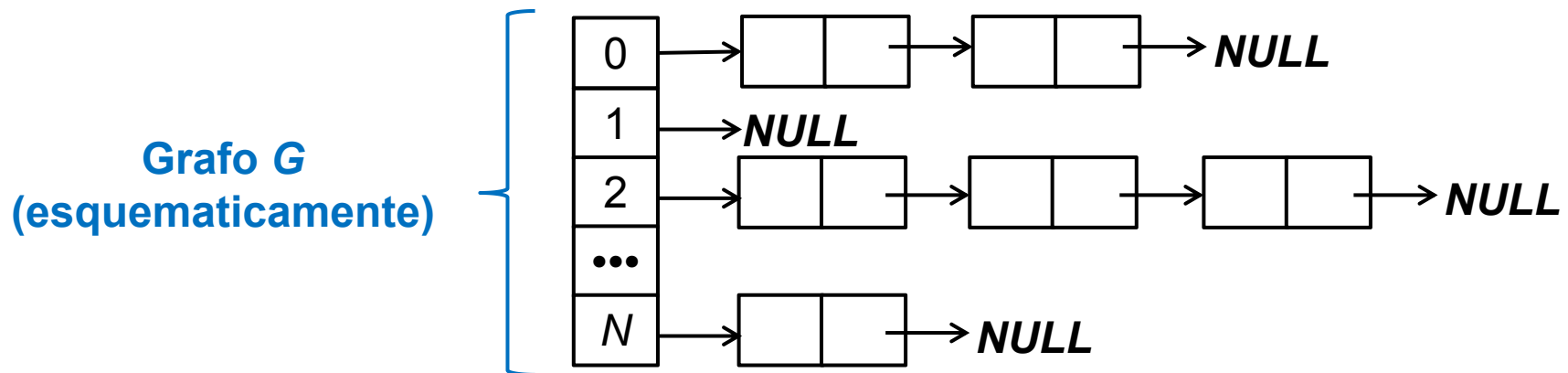


Grafos: listas de adjacência

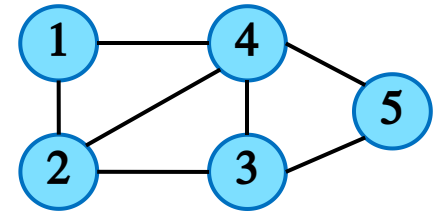
Um **grafo** G com N vértices é representado através de um **vetor** de N posições, onde:

Cada posição refere-se a um vértice V_i

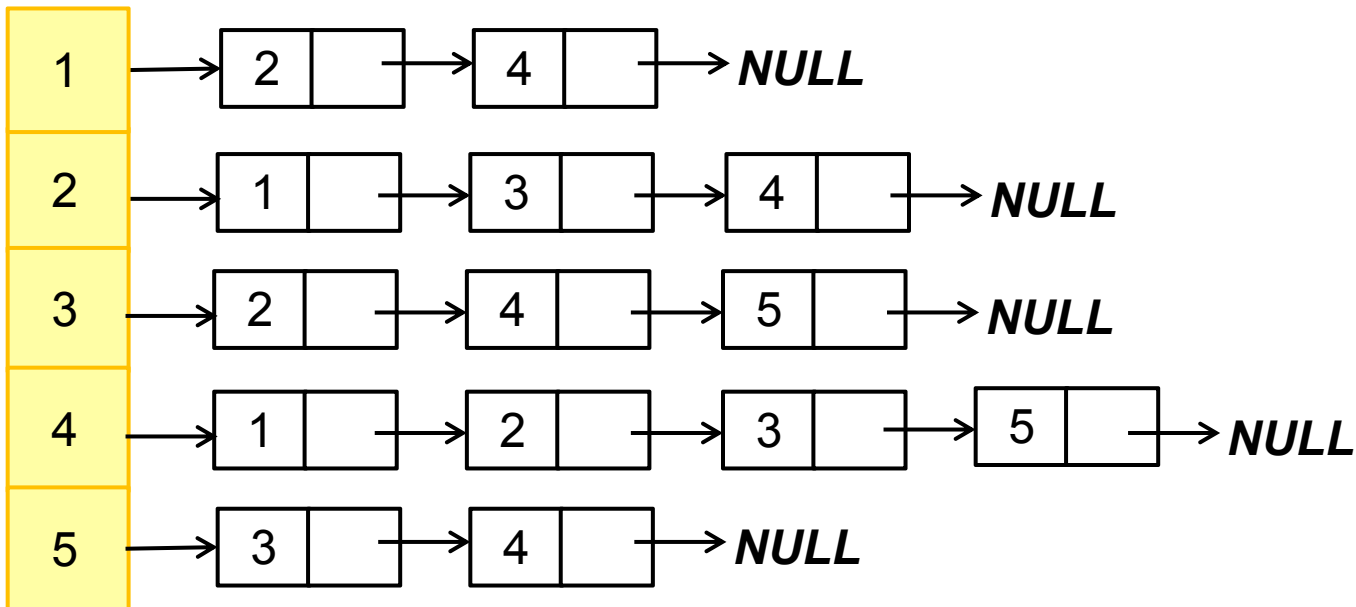
Aponta para uma **lista encadeada** de vértices adjacentes a V_i



Exemplo de listas de adjacência

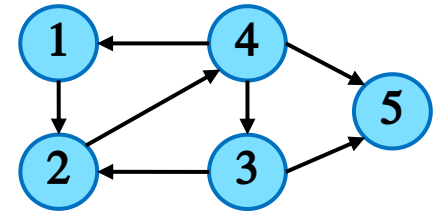


Listas de adjacência

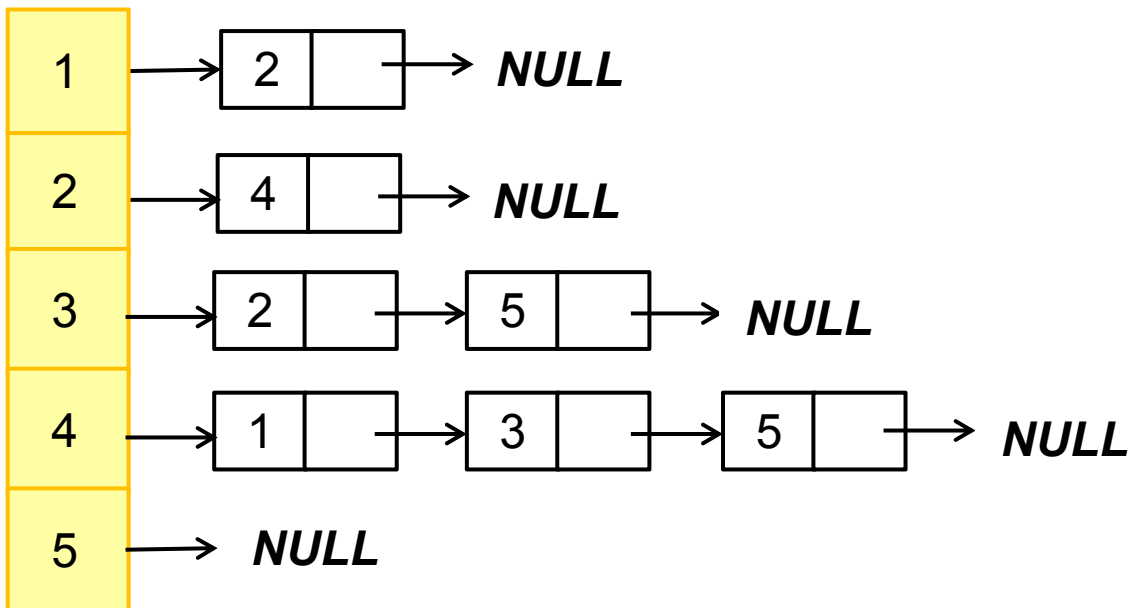


Grafo não direcionado

Exemplo de listas de adjacência



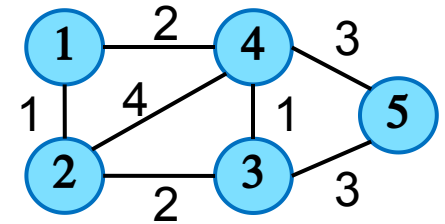
Listas de adjacência



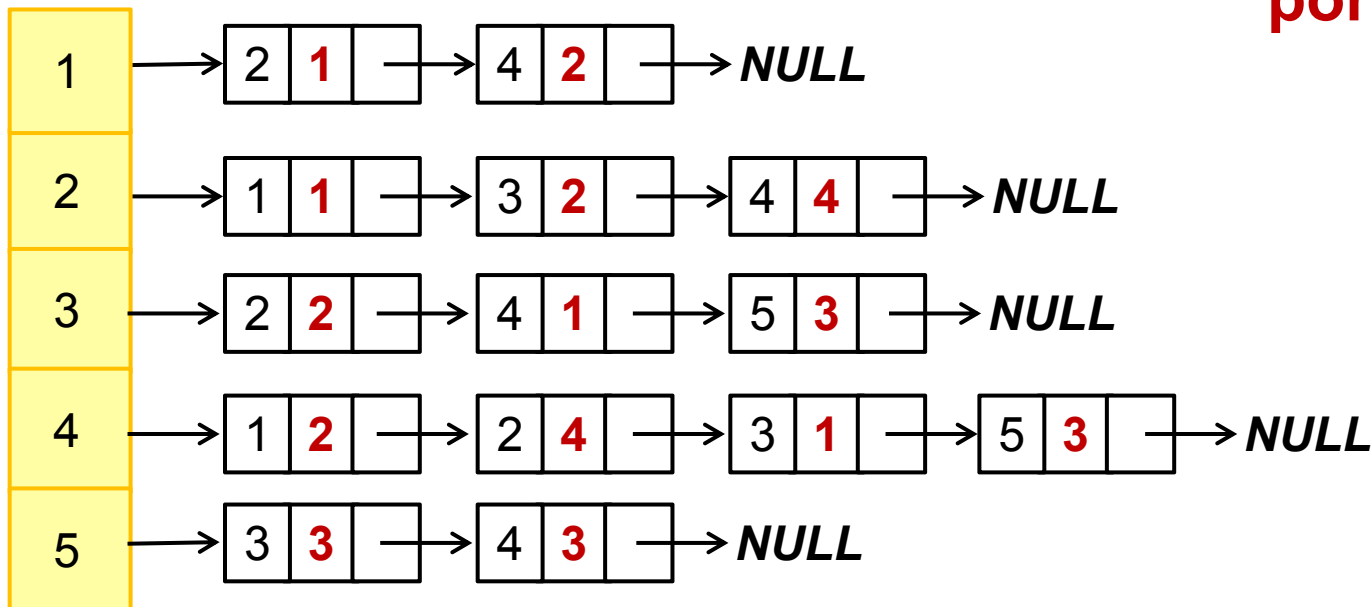
Grafo direcionado

Exemplo de listas de adjacência

Estrutura do nó da lista é modificada para guardar o **peso da aresta**



Listas de adjacência



Grafo não direcionado **ponderado**

Grafos: listas de adjacência

Redundância é mantida na representação de grafos não direcionados

Facilita o acesso às informações

Grafos: listas de adjacência

Redundância é mantida na representação de grafos não direcionados

Facilita o acesso às informações

Representação **mais compacta**

Aloca espaço apenas para arestas existentes

Lista necessita de $O(|V|+|A|)$ de espaço

Indicada para **grafos esparsos** ($|A|$ muito menor que $|V|^2$)

Grafos: listas de adjacência

Redundância é mantida na representação de grafos não direcionados

Facilita o acesso às informações

Representação **mais compacta**

Aloca espaço apenas para arestas existentes

Lista necessita de $O(|V|+|A|)$ de espaço

Indicada para **grafos esparsos** ($|A|$ muito menor que $|V|^2$)

O tempo para determinar se existe uma aresta entre 2 vértices é $O(|V|)$

Grafos: listas de adjacência

Redundância é mantida na representação de grafos não direcionados

Facilita o acesso às informações

Representação **mais compacta**

Aloca espaço apenas para arestas existentes

Lista necessita de $O(|V|+|A|)$ de espaço

Indicada para **grafos esparsos** ($|A|$ muito menor que $|V|^2$)

O tempo para determinar se existe uma aresta entre 2 vértices é $O(|V|)$

Inserção e remoção em listas são **mais complexas**

Grafos: outras formas de representação

Na **matriz de incidência** um grafo com N vértices e M arestas é representado através de uma **matriz $N \times M$**

Grafos: outras formas de representação

Na **matriz de incidência** um grafo com N vértices e M arestas é representado através de uma **matriz $N \times M$**

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se a aresta A_j é incidente ao vértice V_i

$a_{ij} = 0$, caso contrário

Grafos: outras formas de representação

Na **matriz de incidência** um grafo com N vértices e M arestas é representado através de uma **matriz $N \times M$**

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se a aresta A_j é incidente ao vértice V_i

$a_{ij} = 0$, caso contrário

Cada coluna possui apenas dois 1's, referentes aos vértices conectados pela aresta

Grafos: outras formas de representação

Na **matriz de incidência** um grafo com N vértices e M arestas é representado através de uma **matriz $N \times M$**

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se a aresta A_j é incidente ao vértice V_i

$a_{ij} = 0$, caso contrário

Cada coluna possui apenas dois 1's, referentes aos vértices conectados pela aresta

No **grafo ponderado**, o elemento a_{ij} recebe o peso associado com a aresta entre os vértices V_i e V_j

Grafos: outras formas de representação

Na **matriz de incidência** um grafo com N vértices e M arestas é representado através de uma **matriz $N \times M$**

O valor de cada elemento a_{ij} é dado por:

$a_{ij} = 1$, se a aresta A_j é incidente ao vértice V_i

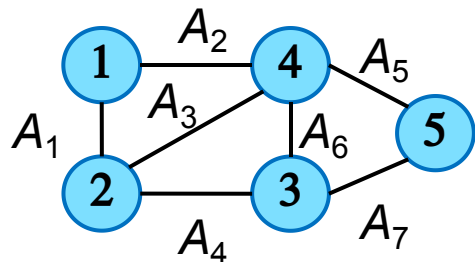
$a_{ij} = 0$, caso contrário

Cada coluna possui apenas dois 1's, referentes aos vértices conectados pela aresta

No **grafo ponderado**, o elemento a_{ij} recebe o peso associado com a aresta entre os vértices V_i e V_j

No **grafo direcionado**, o vértice de destino (V_j) recebe o valor da aresta (P) e o vértice de origem (V_i) recebe $-P$

Exemplo de matriz de incidência



Grafo não direcionado

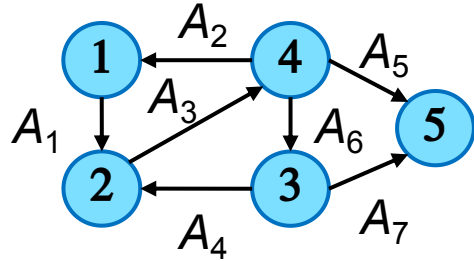
vértices

arestas

	1	2	3	4	5	6	7
1	1	1	0	0	0	0	0
2	1	0	1	1	0	0	0
3	0	0	0	1	0	1	1
4	0	1	1	0	1	1	0
5	0	0	0	0	1	0	1

Matriz de incidência

Exemplo de matriz de incidência



Grafo direcionado

vértices

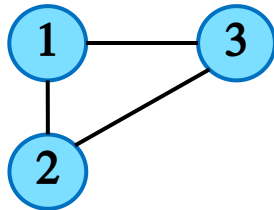
arestas

	1	2	3	4	5	6	7
1	-1	1	0	0	0	0	0
2	1	0	-1	1	0	0	0
3	0	0	0	-1	0	1	-1
4	0	-1	1	0	-1	-1	0
5	0	0	0	0	1	0	1

Matriz de incidência

Grafos: outras formas de representação

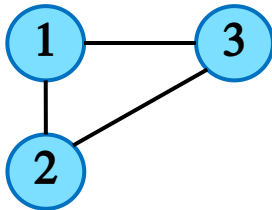
Um grafo é representado pelos seus **conjuntos V** e **A** através de **listas encadeadas**



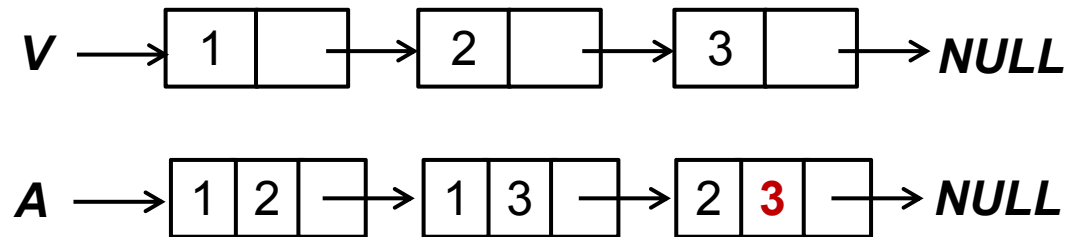
Grafo não direcionado

Grafos: outras formas de representação

Um grafo é representado pelos seus **conjuntos** V e A através de **listas encadeadas**



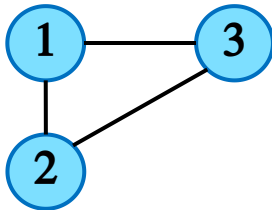
Grafo não direcionado



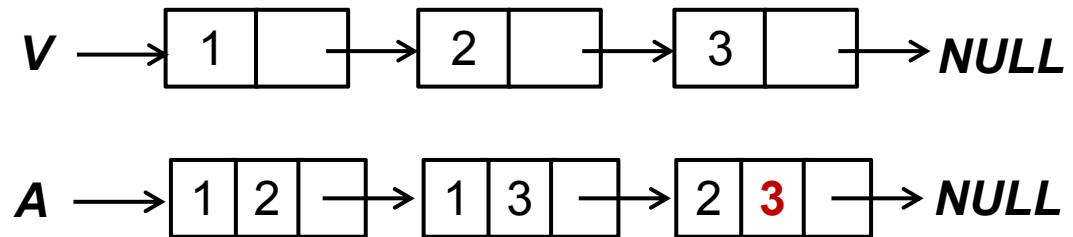
Conjuntos

Grafos: outras formas de representação

Um grafo é representado pelos seus **conjuntos** V e A através de **listas encadeadas**



Grafo não direcionado



Conjuntos

No **grafo ponderado**, o nó da lista de arestas deve incluir o campo peso

Grafos: complexidade das representações

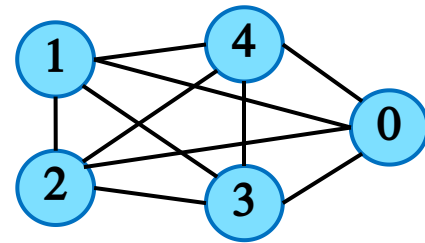
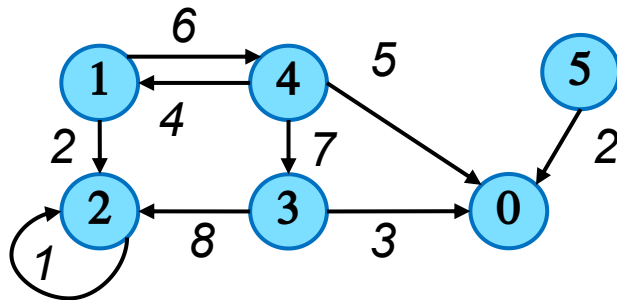
Considere um grafo G com N vértices e M arestas:

M varia de 1 até $N(N-1)/2$

Operação	Matriz de Incidência	Matriz de Adjacência	Listas de Adjacência	Conjuntos
Espaço memória	$O(MN)$	$O(N^2)$	$O(M+N)$	$O(M+N)$
Buscar todos os vizinhos	$O(MN)$	$O(N)$	$O(N)$	$O(M)$
Verificar adjacência V_i e V_j	$O(M)$	$O(1)$	$O(N)$	$O(M)$
Visitar todas as arestas	$O(MN)$	$O(N^2)$	$O(M)$	$O(M)$
Calcular grau de vértice	$O(M)$	$O(N)$	$O(N)$	$O(M)$

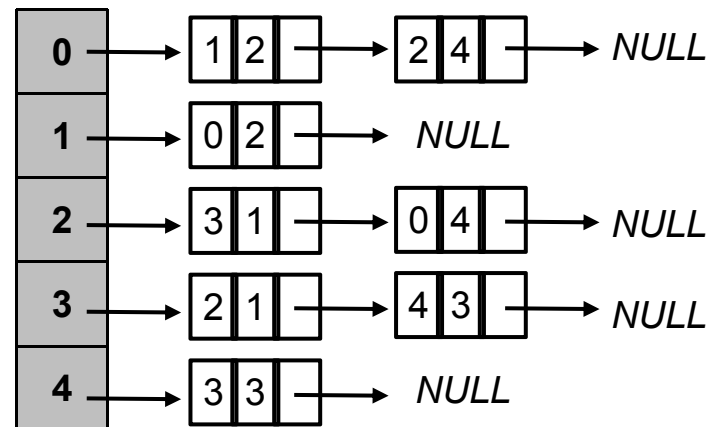
Exercícios

1. Represente os grafos a seguir através da matriz de adjacências, de listas de adjacências e da matriz de incidências.



2. Construa graficamente os grafos que correspondem às representações a seguir, definindo se são direcionados ou não.

	0	1	2	3
0	0	1	0	1
1	0	1	1	0
2	1	0	1	1
3	0	1	1	0



Bibliografia

Slides adaptados do material do Prof. Dr. Marcelo K. Albertini e da Profa. Dra. Denise Guliato.

BACKES, A. Linguagem C Descomplicada: portal de vídeo-aulas para estudo de programação. Disponível em:

<https://programacaodescomplicada.wordpress.com/indice/estrutura-de-dados/>

CORMEN, T.H. et al. Algoritmos: Teoria e Prática, Campus, 2002

ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C (2ª ed.), Thomson, 2004

MORAES, C.R. Estruturas de Dados e Algoritmos: uma abordagem didática (2ª ed.), Futura, 2003