SIN5013
Análise de Algoritmos e Estruturas de Dados

Grafos

Prof. Luciano Antonio Digiampietri
(baseado no material do prof. Norton Trevisan Roman)

Grafos – Conceitos Básicos

O que é um grafo?

Grafos – Conceitos Básicos

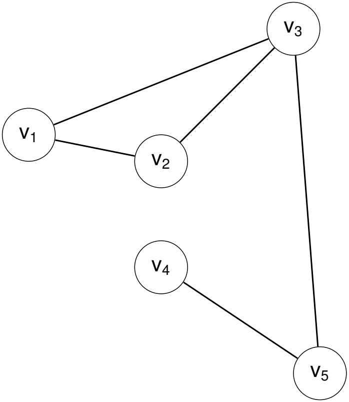
O que é um grafo?

Grafos são estruturas matemáticas (ou modelos matemáticos) que permitem codificar relacionamentos entre pares de objetos

- Os objetos são os **vértices** do grafo
- Os relacionamentos são suas **arestas**

Grafos – Conceitos Básicos

São representados como um conjunto de nós (**vértices**) conectados par a par por linhas (**arestas**)



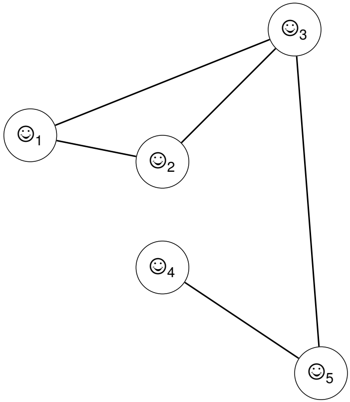
Grafos – Conceitos Básicos

Podem ser utilizados para representar uma infinidade de situações/problemas

Grafos – Conceitos Básicos

Podem ser utilizados para representar uma infinidade de situações/problemas

Podem modelar conexões em redes sociais

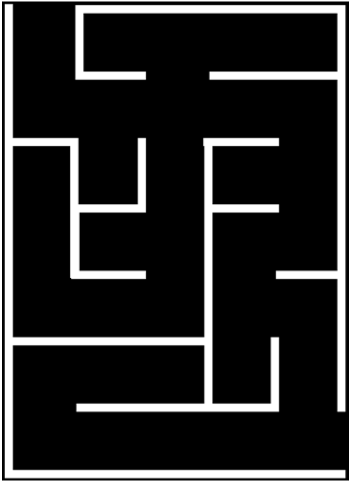


Grafos – Conceitos Básicos

Podem ser utilizados para representar uma infinidade de situações/problemas

Podem modelar conexões em redes sociais

Labirintos



Grafos – Conceitos Básicos

Podem ser utilizados para representar uma infinidade de situações/problemas

Podem modelar conexões em redes sociais

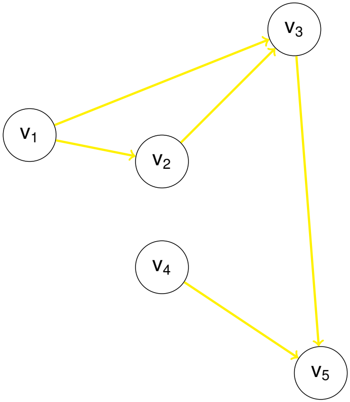
Labirintos

Rotas de metrô

Grafos – Conceitos Básicos

Alguns grafos são **dirigidos** (ou **direcionados**)

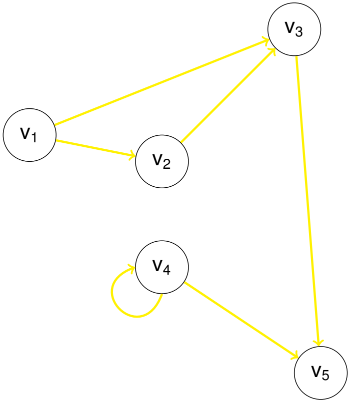
- As relações representadas pelas arestas têm sentido definido
- As arestas só podem ser seguidas em uma única direção



Grafos – Conceitos Básicos

Em **grafos dirigidos**, as arestas são **pares ordenados** de vértices

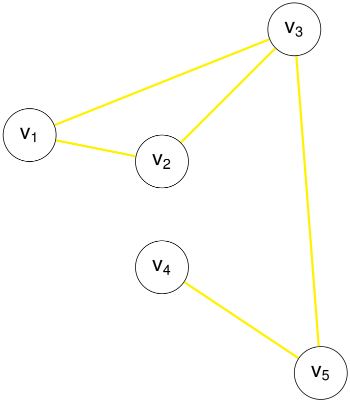
- Saindo de um em direção ao outro
- Mesmo que ambos sejam o mesmo vértice (auto-laços - *self-loop*)



Grafos – Conceitos Básicos

Outros são **não dirigidos** (ou **não direcionados**)

- As relações representadas pelas arestas não têm sentido definido
- As arestas podem ser seguidas em qualquer direção

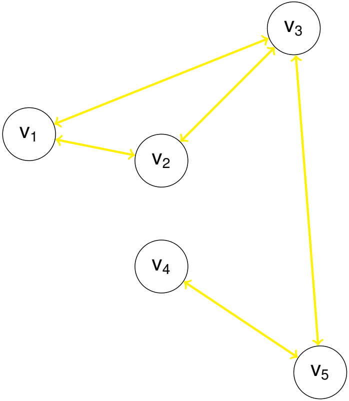


Grafos – Conceitos Básicos

Podemos pensar num grafo não dirigido como um grafo dirigido com arestas de sentido duplo

As arestas são **pares não ordenados** de vértices

Self-loops não são permitidos

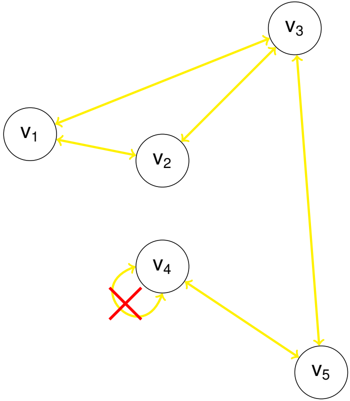


Grafos – Conceitos Básicos

Podemos pensar num grafo não dirigido como um grafo dirigido com arestas de sentido duplo

As arestas são **pares não ordenados** de vértices

Self-loops não são permitidos

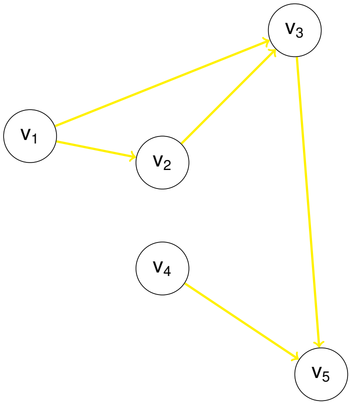


Grafos – Conceitos Básicos

Se (u, v) é uma aresta no grafo, então dizemos que v é **adjacente** a u

- Alternativamente, que v é **vizinho** de u

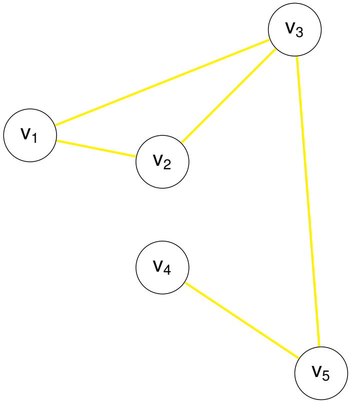
(u, v) significa que a aresta sai de u e entra em v



Grafos – Conceitos Básicos

Em grafos não dirigidos, a relação de adjacência é simétrica

$$(u, v) \Leftrightarrow (v, u)$$



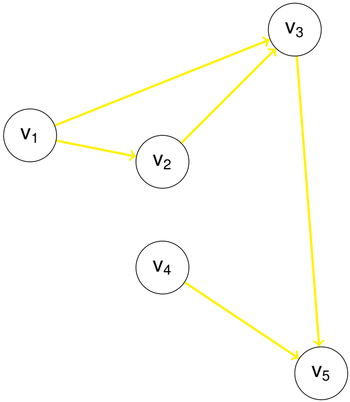
Grafos – Conceitos Básicos

Em grafos não dirigidos, a relação de adjacência é simétrica

$(u, v) \Leftrightarrow (v, u)$

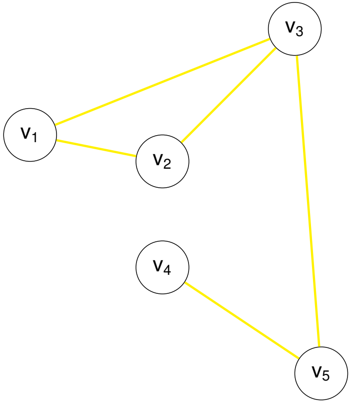
Já em dirigidos, não necessariamente há tal simetria

Há (v_1, v_2), mas não (v_2, v_1)



Grafos – Conceitos Básicos

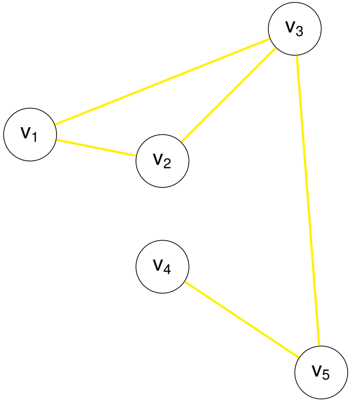
Em grafos não dirigidos, o **grau de um vértice** é o número de arestas que incidem nele



Grafos – Conceitos Básicos

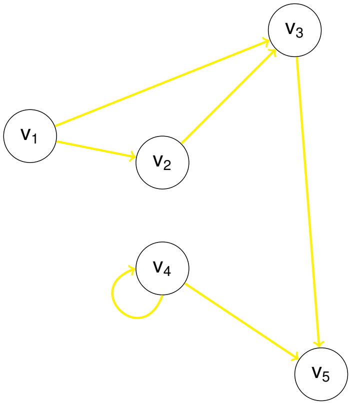
Em grafos não dirigidos, o **grau de um vértice** é o número de arestas que incidem nele

$$gr(v_1) = gr(v_2) = gr(v_5) = 2$$
$$gr(v_3) = 3$$
$$gr(v_4) = 1$$



Grafos – Conceitos Básicos

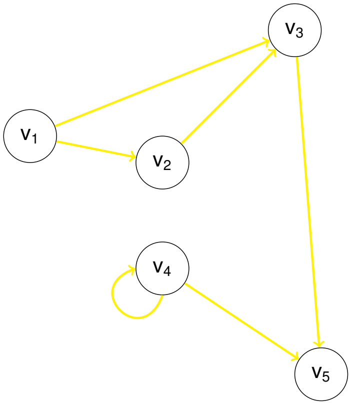
Já em grafos dirigidos, o grau de um vértice é o número de arestas que **saem** do vértice mais o número de arestas que **chegam** nele



Grafos – Conceitos Básicos

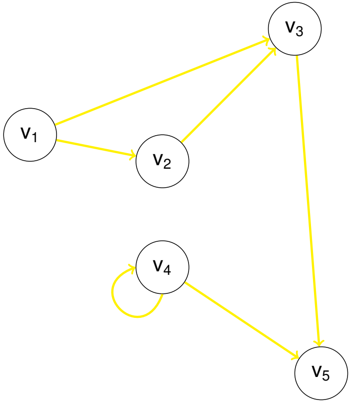
Já em grafos dirigidos, o grau de um vértice é o número de arestas que **saem** do vértice mais o número de arestas que **chegam** nele

$$gr(v_1) = gr(v_2) = gr(v_5) = 2$$
$$gr(v_3) = gr(v_4) = 3$$



Grafos – Conceitos Básicos

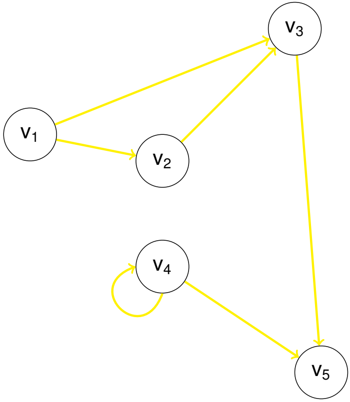
No caso de grafos dirigidos, há dois tipos de graus de vértice:



Grafos – Conceitos Básicos

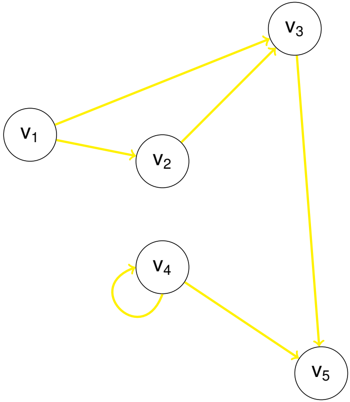
No caso de grafos dirigidos, há dois tipos de graus de vértice:

- **Grau de saída**: número de arestas que saem do vértice
- **Grau de entrada**: número de arestas que chegam no vértice



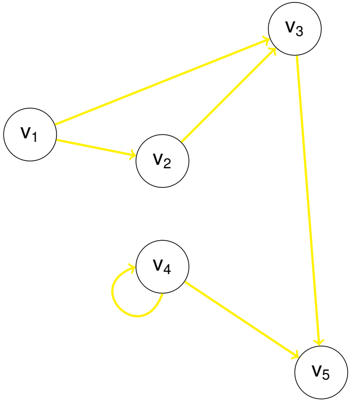
Grafos – Conceitos Básicos

Um **caminho** de um vértice *x* a um vértice *y* é uma sequência de vértices em que, para cada vértice, do primeiro ao penúltimo, há uma aresta ligando esse vértice ao próximo na sequência.



Grafos – Conceitos Básicos

No caso ao lado, alguns caminhos são:



Grafos – Conceitos Básicos

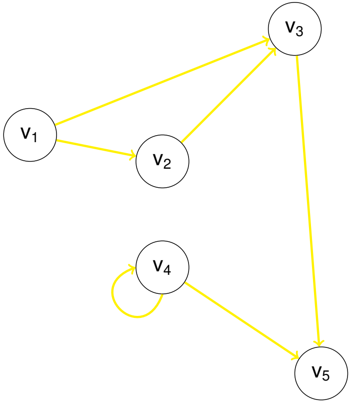
No caso ao lado, alguns caminhos são:

(v_1, v_2, v_3, v_5)

(v_4, v_5)

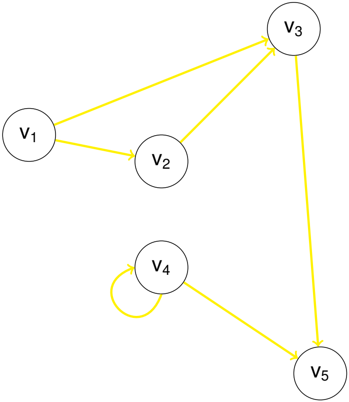
(v_1, v_2, v_3)

(v_4, v_4, v_5)



Grafos – Conceitos Básicos

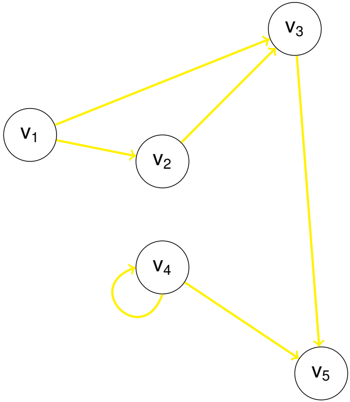
O **comprimento** de um caminho é o número de **arestas** nele



Grafos – Conceitos Básicos

O **comprimento** de um caminho é o número de **arestas** nele

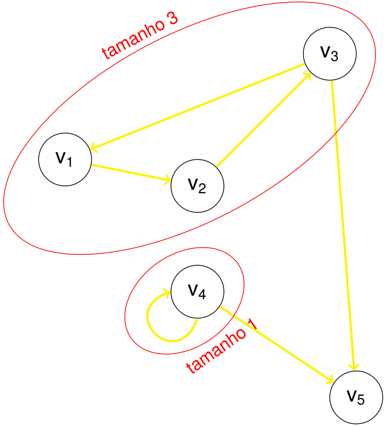
$$compr(v_1, v_2, v_3, v_5) = 3$$
$$compr(v_4, v_5) = 1$$
$$compr(v_1, v_2, v_3) = 2$$
$$compr(v_4, v_4, v_5) = 2$$



Grafos – Conceitos Básicos

Um **ciclo** acontece quando, a partir de um determinado vértice, pudermos percorrer algum caminho que nos leve a esse mesmo vértice

- Em grafos dirigidos, o caminho deve conter pelo menos uma aresta

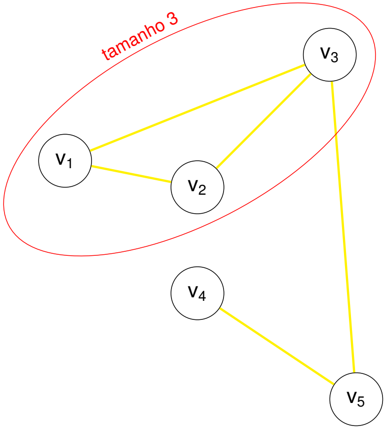


Grafos – Conceitos Básicos

Em grafos não dirigidos, um ciclo deve conter pelo menos 3 arestas

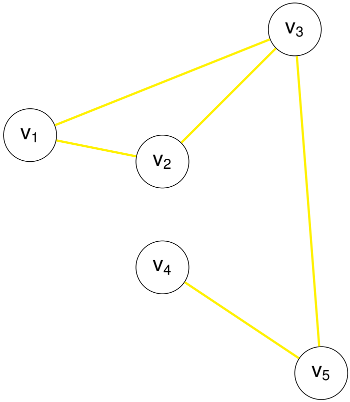
Grafos em que há ao menos um ciclo são chamados de **cíclicos**

Grafos em que não há ciclos são chamados de **acíclicos**



Grafos – Conceitos Básicos

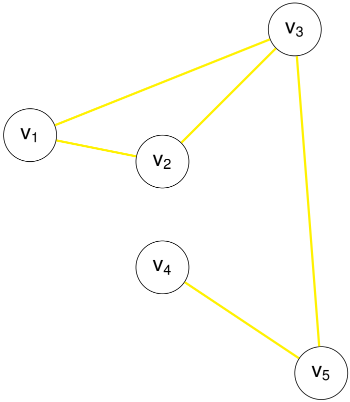
Um grafo não direcionado é **conexo** (ou **conectado**) se cada par de vértices nele estiver conectado por um caminho



Grafos – Conceitos Básicos

Um grafo não direcionado é **conexo** (ou **conectado**) se cada par de vértices nele estiver conectado por um caminho

O grafo ao lado é conexo

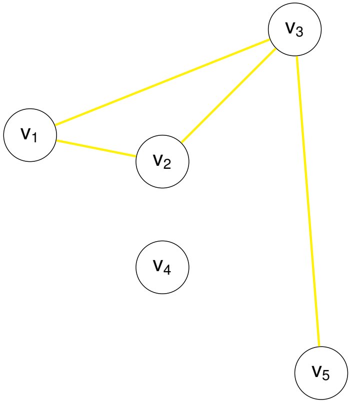


Grafos – Conceitos Básicos

Um grafo não direcionado é **conexo** (ou **conectado**) se cada par de vértices nele estiver conectado por um caminho

O grafo ao lado é conexo

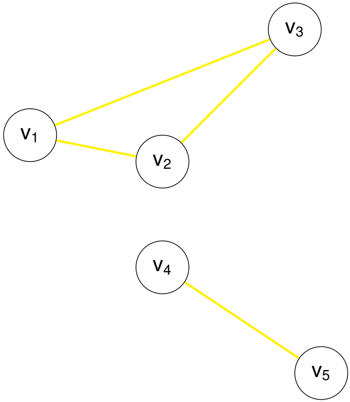
Agora é **desconexo**



Grafos – Conceitos Básicos

Um grafo não direcionado é **conexo** (ou **conectado**) se cada par de vértices nele estiver conectado por um caminho

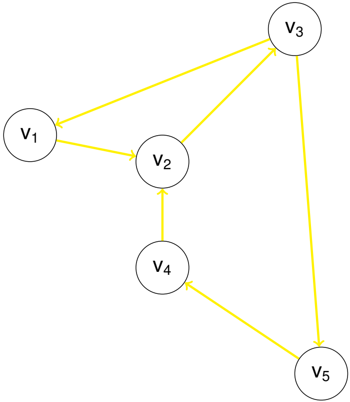
- O grafo ao lado é conexo
- Agora é **desconexo**
- E continua assim



Grafos – Conceitos Básicos

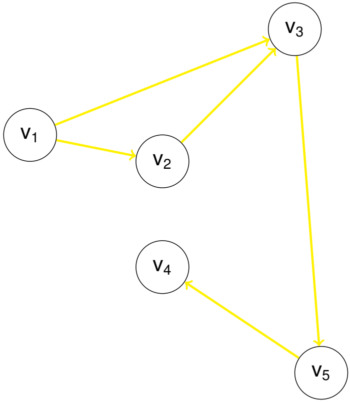
Um grafo dirigido é **fortemente conexo** se existir um caminho entre qualquer par de vértices no grafo

- Contém um caminho direto de *u* para *v* e um caminho direto *v* para *u* para cada par de vértices (*u*, *v*)



Grafos – Conceitos Básicos

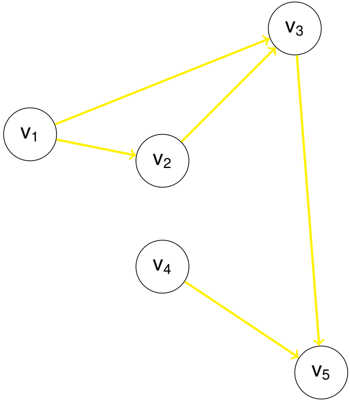
Um grafo dirigido é **conexo** se possuir um caminho de *u* para *v*, ou um caminho de *v* para *u*, para cada par de vértices (u, v)



Grafos – Conceitos Básicos

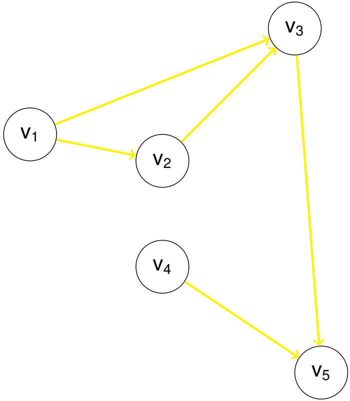
Um grafo dirigido é **fracamente conexo** se a substituição de todas as suas arestas por arestas **não-direcionadas** produz um grafo conexo.

Ex: não há caminho de $v_4 \rightarrow v_3$ nem de $v_3 \rightarrow v_4$



Grafos – Conceitos Básicos

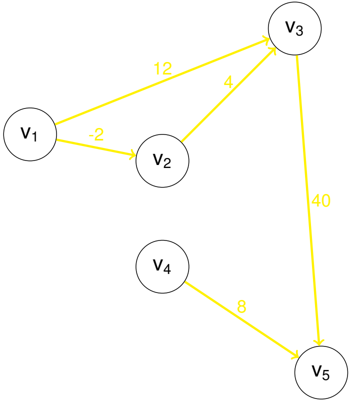
Grafos também podem ser **ponderados**



Grafos – Conceitos Básicos

Grafos também podem ser **ponderados**

- Caso em que possuem pesos associados às suas arestas
- Esses pesos podem representar custos, distâncias etc.



Grafos – Conceitos Básicos

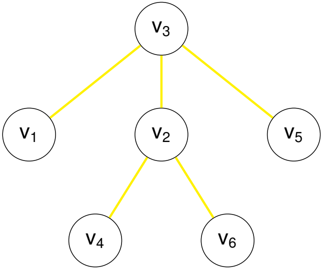
Por fim, temos um tipo já conhecido de grafo...

Grafos – Conceitos Básicos

Por fim, temos um tipo já conhecido de grafo...

A **árvore**

- Grafo acíclico
- Conexo
- Não-dirigido



Grafos – Representação

Como podemos representar um grafo?

Grafos – Representação

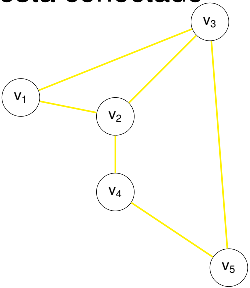
Como podemos representar um grafo?

- Como um mapeamento de cada nó à lista de nós aos quais ele está conectado

Grafos – Representação

Como podemos representar um grafo?

- Como um mapeamento de cada nó à lista de nós aos quais ele está conectado

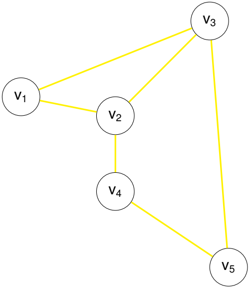


Não dirigido

Grafos – Representação

Como podemos representar um grafo?

- Como um mapeamento de cada nó à lista de nós aos quais ele está conectado



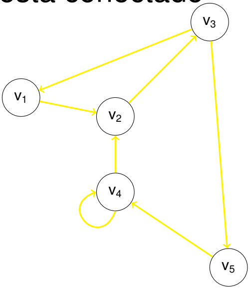
Não dirigido

Nó	*Conectado a*
v_1	v_2, v_3
v_2	v_1, v_3, v_4
v_3	v_1, v_2, v_5
v_4	v_2, v_5
v_5	v_3, v_4

Grafos – Representação

Como podemos representar um grafo?

- Como um mapeamento de cada nó à lista de nós aos quais ele está conectado

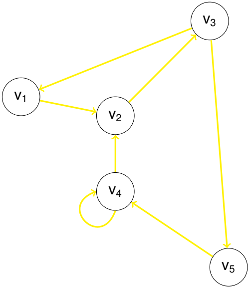


Dirigido

Grafos – Representação

Como podemos representar um grafo?

Como um mapeamento de cada nó à lista de nós aos quais ele está conectado



Dirigido

Nó	*Conectado a*
v_1	v_2
v_2	v_3
v_3	v_1, v_5
v_4	v_2, v_4
v_5	v_4

Grafos – Representação

Representação computacional de grafos

Grafos – Representação

Representação computacional de grafos

Existem duas maneiras usuais de representar grafos:

- Matrizes de adjacência
- Listas de adjacência

Grafos – Matrizes de Adjacências

Uma **matriz de adjacências** A de um grafo com n vértices é uma matriz $n \times n$ de bits, em que:

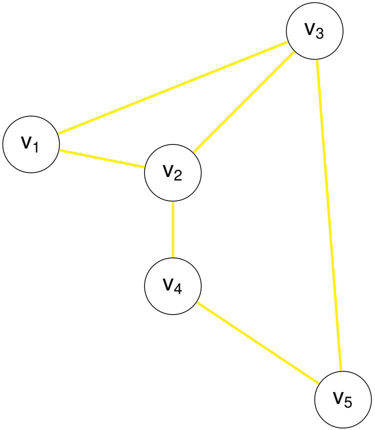
Grafos – Matrizes de Adjacências

Uma **matriz de adjacências** A de um grafo com n vértices é uma matriz $n \times n$ de bits, em que:

$A[i, j] = 1$ se houver uma aresta indo do vértice i para o vértice j no grafo.

$A[i, j] = 0$ se não houver aresta indo de i para j

Grafos – Matrizes de Adjacências



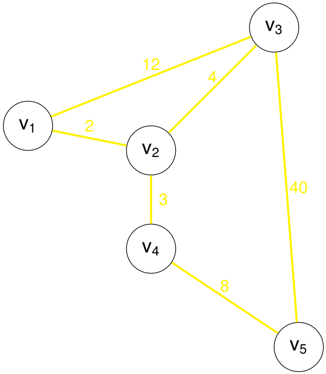
	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	0	0
v_2	1	0	1	1	0
v_3	1	1	0	0	1
v_4	0	1	0	0	1
v_5	0	0	1	1	0

Grafos – Matrizes de Adjacências

E se o grafo for ponderado?

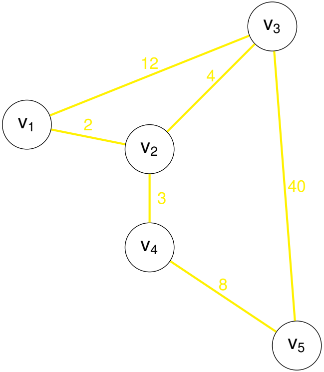
Grafos – Matrizes de Adjacências

E se o grafo for ponderado?



Grafos – Matrizes de Adjacências

E se o grafo for ponderado? Não usamos bits.



	v_1	v_2	v_3	v_4	v_5
v_1	0	2	12	0	0
v_2	2	0	4	3	0
v_3	12	4	0	0	40
v_4	0	3	0	0	8
v_5	0	0	40	8	0

Grafos – Matrizes de Adjacências

E se o grafo for ponderado? Não usamos bits.

Cuidado!

	v_1	v_2	v_3	v_4	v_5
v_1	0	2	12	0	0
v_2	2	0	4	3	0
v_3	12	4	0	0	40
v_4	0	3	0	0	8
v_5	0	0	40	8	0

Grafos – Matrizes de Adjacências

E se o grafo for ponderado? Não usamos bits.

Cuidado!

Como diferenciar não aresta de um valor válido (como 0)?

	v_1	v_2	v_3	v_4	v_5
v_1	**0**	2	12	**0**	**0**
v_2	2	**0**	4	3	**0**
v_3	12	4	**0**	**0**	40
v_4	**0**	3	**0**	**0**	8
v_5	**0**	**0**	40	8	**0**

Grafos – Matrizes de Adjacências

E se o grafo for ponderado? Não usamos bits.

Cuidado!

Como diferenciar não aresta de um valor válido (como 0)?

Deve-se definir um padrão

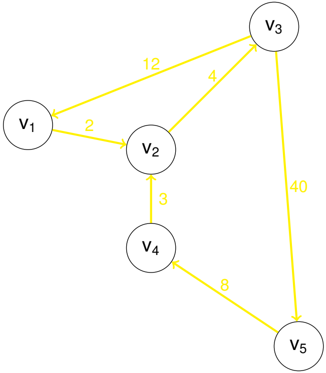
	v_1	v_2	v_3	v_4	v_5
v_1	**0**	2	12	**0**	**0**
v_2	2	**0**	4	3	**0**
v_3	12	4	**0**	**0**	40
v_4	**0**	3	**0**	**0**	8
v_5	**0**	**0**	40	8	**0**

Grafos – Matrizes de Adjacências

E se o grafo for dirigido?

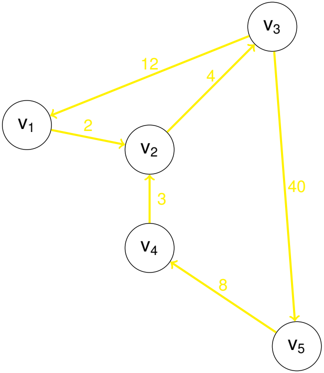
Grafos – Matrizes de Adjacências

E se o grafo for dirigido?



Grafos – Matrizes de Adjacências

E se o grafo for dirigido?



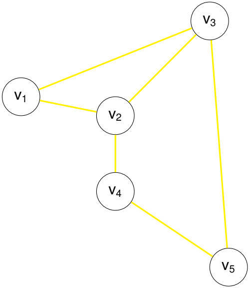
	v_1	v_2	v_3	v_4	v_5
v_1	0	2	0	0	0
v_2	0	0	4	0	0
v_3	12	0	0	0	40
v_4	0	3	0	0	0
v_5	0	0	0	8	0

Grafos – Listas de Adjacências

A segunda maneira de representar um grafo é por meio de uma lista de adjacências

Grafos – Listas de Adjacências

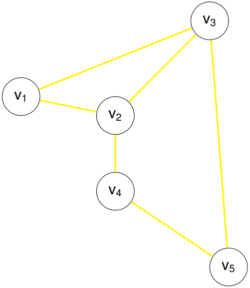
A segunda maneira de representar um grafo é por meio de uma lista de adjacências



Não dirigido

Grafos – Listas de Adjacências

A segunda maneira de representar um grafo é por meio de uma lista de adjacências

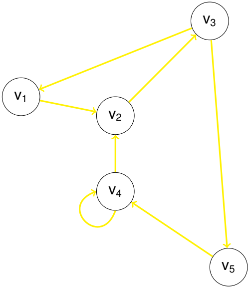


Não dirigido

Nó	*Conectado a*
v_1	v_2, v_3
v_2	v_1, v_3, v_4
v_3	v_1, v_2, v_5
v_4	v_2, v_5
v_5	v_3, v_4

Grafos – Listas de Adjacências

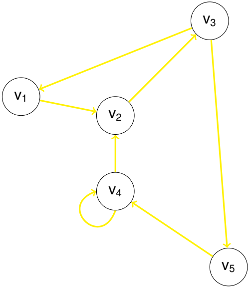
A segunda maneira de representar um grafo é por meio de uma lista de adjacências



Dirigido

Grafos – Listas de Adjacências

A segunda maneira de representar um grafo é por meio de uma lista de adjacências



Dirigido

Nó	*Conectado a*
v_1	v_2
v_2	v_3
v_3	v_1 , v_5
v_4	v_2 , v_4
v_5	v_4

Grafos – Listas de Adjacências

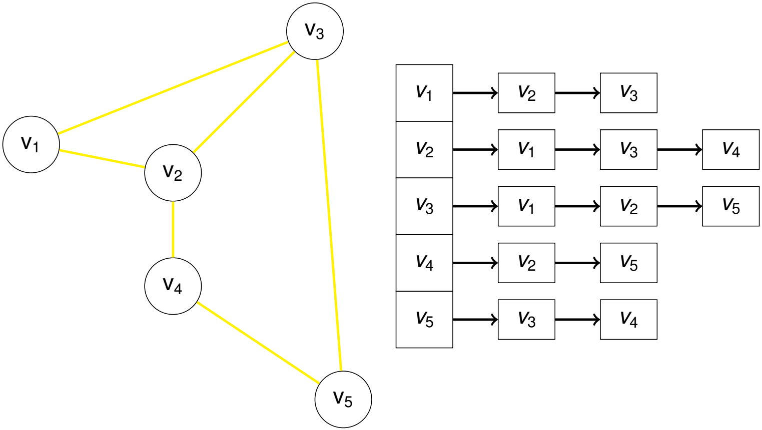
Uma **lista de adjacências** de um grafo com *n* vértices consiste de um arranjo de *n* listas ligadas, uma para cada vértice no grafo.

Grafos – Listas de Adjacências

Uma **lista de adjacências** de um grafo com *n* vértices consiste de um arranjo de *n* listas ligadas, uma para cada vértice no grafo.

- Para cada vértice u, a lista contém todos os vizinhos de u
- Ou seja, todos os vértices v_i para os quais existe uma aresta (u, v_i)

Grafos – Listas de Adjacências

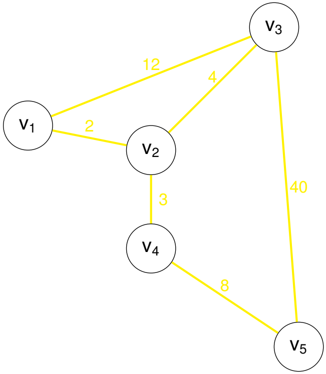


Grafos – Listas de Adjacências

E se o grafo for ponderado?

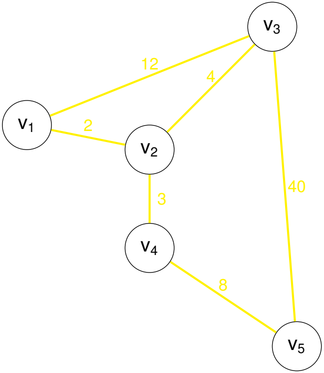
Grafos – Listas de Adjacências

E se o grafo for ponderado?



Grafos – Listas de Adjacências

E se o grafo for ponderado? Armazenamos os pesos nas lista.



$v_1 \rightarrow v_2, 2 \rightarrow v_3, 12$

$v_2 \rightarrow v_1, 2 \rightarrow v_3, 4 \rightarrow v_4, 3$

$v_3 \rightarrow v_1, 12 \rightarrow v_2, 4 \rightarrow v_5, 40$

$v_4 \rightarrow v_2, 3 \rightarrow v_5, 8$

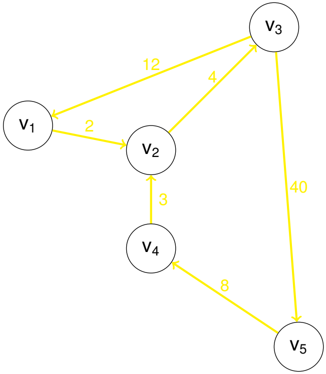
$v_5 \rightarrow v_3, 40 \rightarrow v_4, 8$

Grafos – Listas de Adjacências

E se o grafo for dirigido?

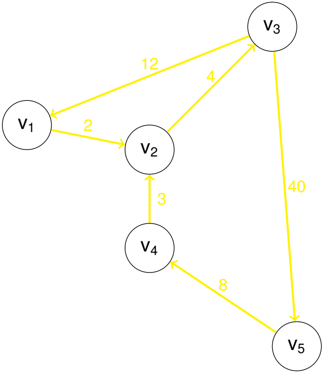
Grafos – Listas de Adjacências

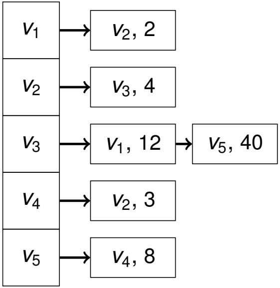
E se o grafo for dirigido?



Grafos – Listas de Adjacências

E se o grafo for dirigido?





Grafos – Representação

E quando usamos uma ou outra representação?

Grafos – Representação

E quando usamos uma ou outra representação?

Vai depender da densidade do grafo

- Se **denso** (quando possui muitas arestas em relação ao número de vértices) ou **esparso** (com poucas arestas)

Vai depender das operações que queremos executar

SIN5013
Análise de Algoritmos e Estruturas de Dados

Grafos

Prof. Luciano Antonio Digiampietri
(baseado no material do prof. Norton Trevisan Roman)