ACH2024 –

Algoritmos e Estruturas de Dados II

Prof. Helton Hideraldo Bíscaro
heltonhb@usp.br
Sala 352 b

Aula 1 –

Introdução sobre a disciplina, Conceitos básicos de grafos

Prof. Helton Hideraldo Bíscaro
heltonhb@usp.br
Sala 352 b

PRÉ - REQUISITO

AED 1

Objetivo e Programa

- Objetivo:

Estudo e resolução de problemas que utilizem estruturas de dados de memória secundária. Estudo e desenvolvimento de algoritmos baseados em grafos.

- Programa:

Algoritmos para classificação externa. Arquivos de organização sequencial e randômica. Consulta e atualização de arquivos. Técnicas de indexação, árvores-B e hashing externo. Estruturas de dados para representação de grafos e seus algoritmos.

- Programação em C, C++

Bibliografia:

- descrita na ementa da disciplina no Júpiter,
- descrita no final de cada aula,
- e:

· ZIVIANI, N. Projetos de Algoritmos - com implementações em Pascal e C. 3ª ed. revista e ampliada Cengage Learning, 2011.

· AHO,A.V.; HOPCROFT,J.E.; ULLMAN,J.D. Data Structure and Algorithms. Readings, Addison Wesley, 1982.

· CORMEN, H.T.; LEISERSON, C.E.; RIVEST, R.L. Introduction to Algorithms, MIT Press, McGraw-Hill, 1999

· TENEMBAUM,A.M. et al Data Structures Using C, Prentice-Hall, 1990.

· WIRTH,N. Algorithms and Data Structures, Englewood Cliffs, Prentice-Hall, 1986.

· DROSDEK, A. Estrutura de Dados e Algoritmos em C++. Cencage Learning, 2002.

· GOODRICH, M.; TAMASSIA, R. "Estruturas de Dados e Algoritmos em Java". Ed. Bookman, 2a. Ed. 2002

· FOLK, M.J; ZOELLICK, B. "File Structures". Addison-Wesley, 2a. Ed. 1991.

Bibliografia

Não haverá listas de exercícios para ENTREGA

Usem todos os livros possíveis (principalmente os recomendados) para fazerem o exercícios como forma de estudo (para a prova, inclusive)

Participação

- Se você não participar da aula perderá o seu tempo!
  - Aprender é diferente de decorar, e para aprender é preciso raciocinar.

Recursos

- E-disciplinas:
  - PDFs das aula no “Repositório” (mas façam anotações em seus cadernos! Os slides NÃO têm tudo. São numerados para vocês os complementarem!)
  - Emails em “Mensagens” (usar sempre “com cópia para o e-mail do destinatário”)

Avaliação

- 2 provas:

  MP = (P1 + P2)/2

- 2 Exercícios programas (EPs) individuais:

  MT = (2*EP1 + 3* EP2)/5

- Média 1ª avaliação (antes da REC) M1:

  Se MP >= 5 E MT >= 5

  M1 = (0.7*MP + 0.3* MT)

  senão M1 = min(MP, MT)

- Prova SUB:
  - só para quem perdeu alguma prova (não precisa de atestado, mas prefira não fazê-la, é mais difícil que as outras)
  - substitui só uma prova (a que faltou; se faltou nas duas substitui a P2)

- RECuperação (Média 2ª avaliação - M2):

  M2 = (M1 + REC)/2

Parte 1 da Disciplina:

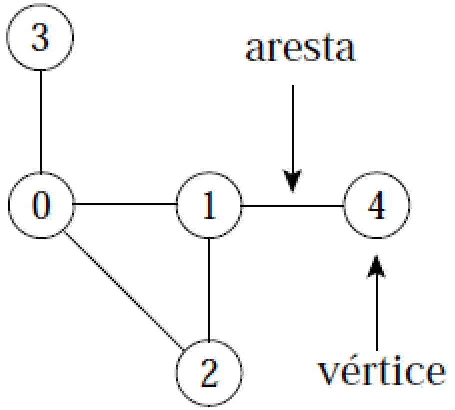
GRAFOS

Aula 1 – Conceitos Básicos

BASEADA NOS SLIDES DO CAP 7 DO LIVRO:

Grafos

- **Grafo**: conjunto de vértices e arestas.
- **Vértice**: objeto simples que pode ter nome e outros atributos.
- **Aresta**: conexão entre dois vértices.



As arestas também podem ter atributos (normalmente um "peso")

- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

Grafos

- Então para que podem servir os grafos?

Grafos - Motivação

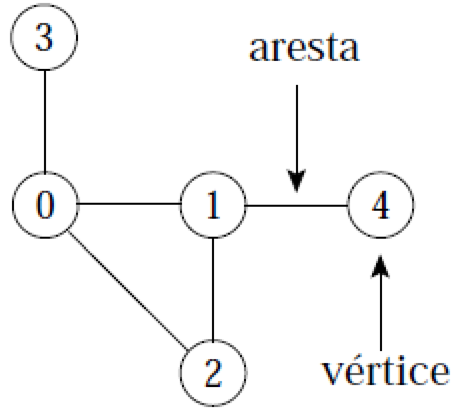
- Muitas aplicações em computação necessitam considerar conjunto de conexões entre pares de objetos:
  - Existe um caminho para ir de um objeto a outro seguindo as conexões?
  - Qual é a menor distância entre um objeto e outro objeto?
  - Quantos outros objetos podem ser alcançados a partir de um determinado objeto?
- Existe um tipo abstrato chamado grafo que é usado para modelar tais situações.

Aplicações

- Alguns exemplos de problemas práticos que podem ser resolvidos através de uma modelagem em grafos:
  - Ajudar máquinas de busca a localizar informação relevante na Web.
  - Descobrir os melhores casamentos entre posições disponíveis em empresas e pessoas que aplicaram para as posições de interesse.
  - Descobrir qual é o roteiro mais curto para visitar as principais cidades de uma região turística.

Retomando...

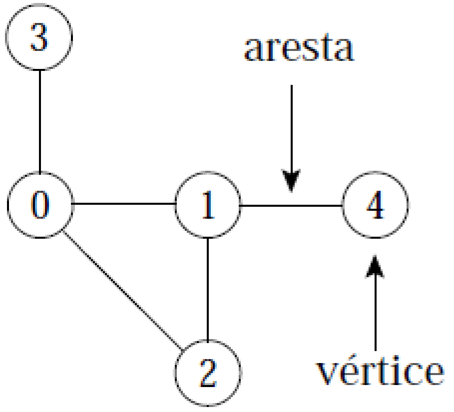
- **Grafo**: conjunto de vértices e arestas.
- **Vértice**: objeto simples que pode ter nome e outros atributos.
- **Aresta**: conexão entre dois vértices.



- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

Retomando...

- **Grafo**: conjunto de vértices e arestas.
- **Vértice**: objeto simples que pode ter nome e outros atributos.
- **Aresta**: conexão entre dois vértices.

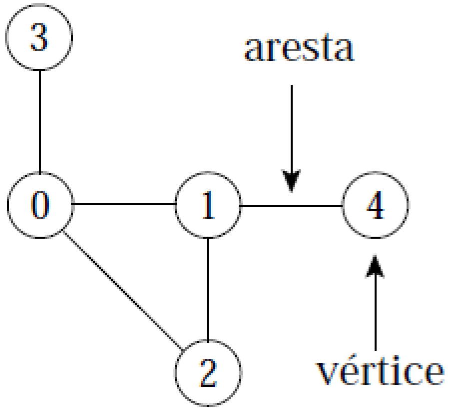


- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

Será que essa aresta, como está aqui, é o suficiente para caracterizar conexões?

Retomando...

- **Grafo**: conjunto de vértices e arestas.
- **Vértice**: objeto simples que pode ter nome e outros atributos.
- **Aresta**: conexão entre dois vértices.



- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

Será que essa aresta, como está aqui, é o suficiente para caracterizar conexões?

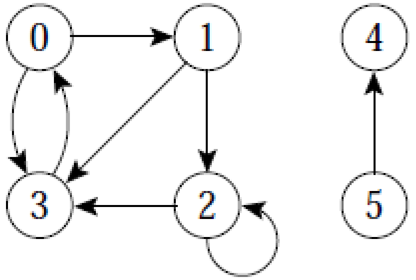
Imagine fazer um caminho pelas ruas das cidades… com mão e contra-mão….

PAR ORDENADO!

Grafos Direcionados

- Um **grafo direcionado** G é um par (V, A), onde V é um conjunto finito de vértices e A é uma relação binária em V.
  - Uma aresta (u, v) sai do vértice u e entra no vértice v. O vértice v é **adjacente** ao vértice u. (adjacente vem depois)
  - Podem existir arestas de um vértice para ele mesmo, chamadas de *self-loops*.

Como descrever?



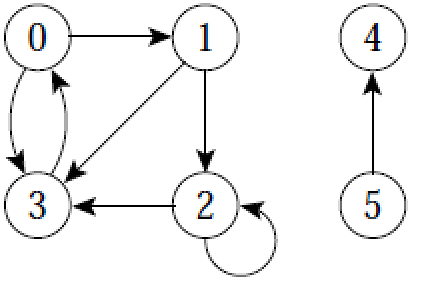
Também chamados **Digrafos** (sem acento)

PAR ORDENADO!

Grafos Direcionados

- Um **grafo direcionado** G é um par (V, A), onde V é um conjunto finito de vértices e A é uma relação binária em V.
  - Uma aresta (u, v) sai do vértice u e entra no vértice v. O vértice v é **adjacente** ao vértice u. (adjacente vem depois)
  - Podem existir arestas de um vértice para ele mesmo, chamadas de *self-loops*.

Como descrever?



(u, u)

Também chamados **Digrafos** (sem acento)

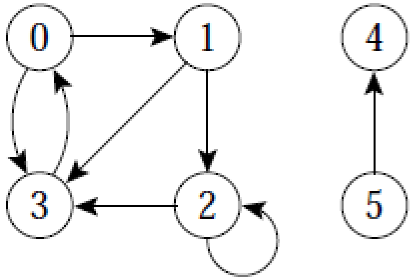
PAR ORDENADO!

Grafos Direcionados

- Um **grafo direcionado** G é um par (V, A), onde V é um conjunto finito de vértices e A é uma relação binária em V.
  - Uma aresta (u, v) sai do vértice u e entra no vértice v. O vértice v é **adjacente** ao vértice u. (adjacente vem depois)
  - Podem existir arestas de um vértice para ele mesmo, chamadas de *self-loops*.

Como descrever?

Descreva matematicamente (formalmente) esse grafo:



(u, u)

Também chamados **Digrafos** (sem acento)

PAR ORDENADO!

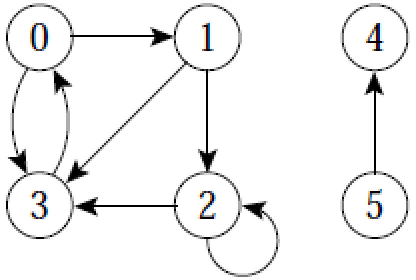
Grafos Direcionados

- Um **grafo direcionado** G é um par (V, A), onde V é um conjunto finito de vértices e A é uma relação binária em V.
  - Uma aresta (u, v) sai do vértice u e entra no vértice v. O vértice v é **adjacente** ao vértice u. (adjacente vem depois)
  - Podem existir arestas de um vértice para ele mesmo, chamadas de *self-loops*.

Como descrever?

Descreva matematicamente
(formalmente) esse grafo:

G = (V,A)
V = {0, 1, 2, 3, 4, 5}
A = {(0,1), (0,3), (1,2), (1,3),
(2,2), (2,3), (3,0), (5,4)}



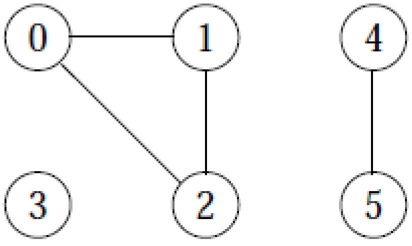
(u, u)

Também chamados **Digrafos** (sem acento)

Grafos Não Direcionados

- Um **grafo não direcionado** G é um par (V, A), onde o conjunto de arestas A é constituído de pares de vértices não ordenados.
  - As arestas (u, v) e (v, u) são consideradas como uma única aresta. A relação de adjacência é simétrica.
  - *Self-loops* não são permitidos.

(apenas uma participa da definição do grafo)

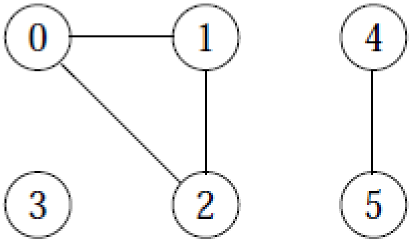


Grafos Não Direcionados

- Um **grafo não direcionado** G é um par (V, A), onde o conjunto de arestas A é constituído de pares de vértices não ordenados.
  - As arestas (u, v) e (v, u) são consideradas como uma única aresta. A relação de adjacência é simétrica.
  - *Self-loops* não são permitidos.

(apenas uma participa da definição do grafo)

Descreva matematicamente (formalmente) esse grafo:



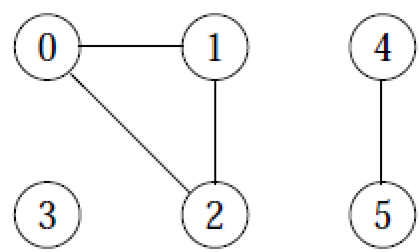
Grafos Não Direcionados

- Um **grafo não direcionado** G é um par (V, A), onde o conjunto de arestas A é constituído de pares de vértices não ordenados.
  - As arestas (u, v) e (v, u) são consideradas como uma única aresta. A relação de adjacência é simétrica.
  - *Self-loops* não são permitidos.

(apenas uma participa da definição do grafo)

Descreva matematicamente (formalmente) esse grafo:

G = (V,A)
V = {0, 1, 2, 3, 4, 5}
A = {(0,1), (1,2), (2,0), (4,5)}



Grafos Não Direcionados

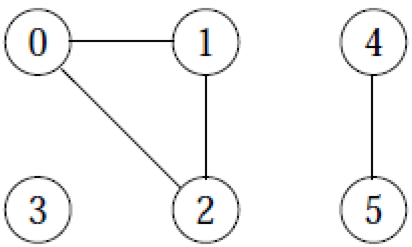
- Um **grafo não direcionado** G é um par (V, A), onde o conjunto de arestas A é constituído de pares de vértices não ordenados.
  - As arestas (u, v) e (v, u) são consideradas como uma única aresta. A relação de adjacência é simétrica.
  - *Self-loops* não são permitidos.

(apenas uma participa da definição do grafo)

Descreva matematicamente (formalmente) esse grafo:

G = (V,A)
V = {0, 1, 2, 3, 4, 5}
A = {(0,1), (1,2), (2,0), (4,5)}

Posso inverter os vértices nas arestas?



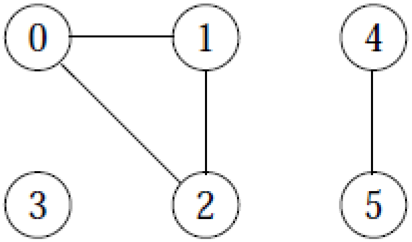
Grafos Não Direcionados

- Um **grafo não direcionado** G é um par (V, A), onde o conjunto de arestas A é constituído de pares de vértices não ordenados.
  - As arestas (u, v) e (v, u) são consideradas como uma única aresta. A relação de adjacência é simétrica.
  - *Self-loops* não são permitidos.

(apenas uma participa da definição do grafo)

Descreva matematicamente (formalmente) esse grafo:

G = (V,A)
V = {0, 1, 2, 3, 4, 5}
A = {(1,0), (1,2), (2,0), (4,5)}



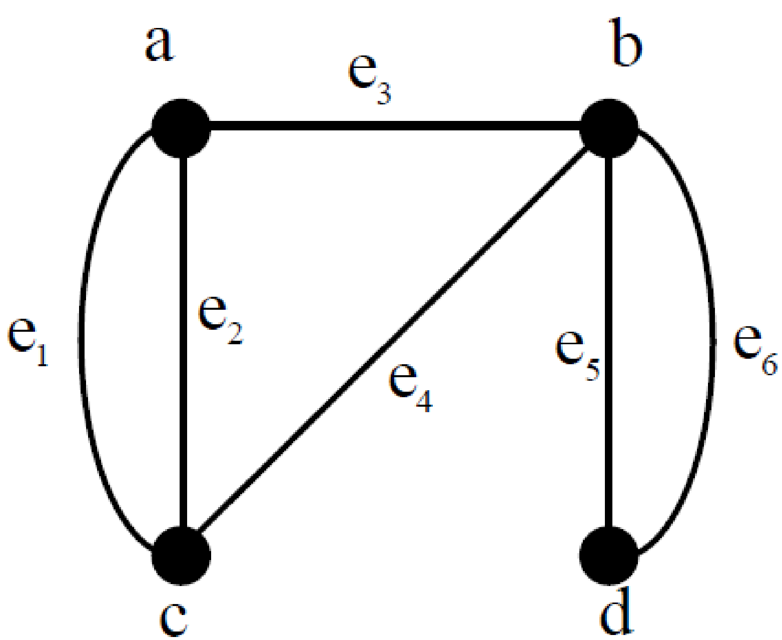
Posso inverter os vértices nas arestas?

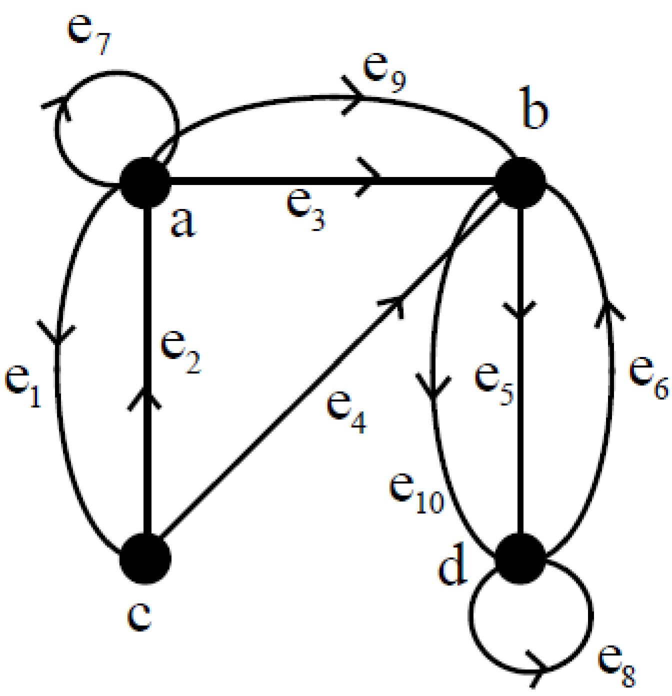
SIM!!!

Observação

Essas definições anteriores referem-se aos grafos "simples"

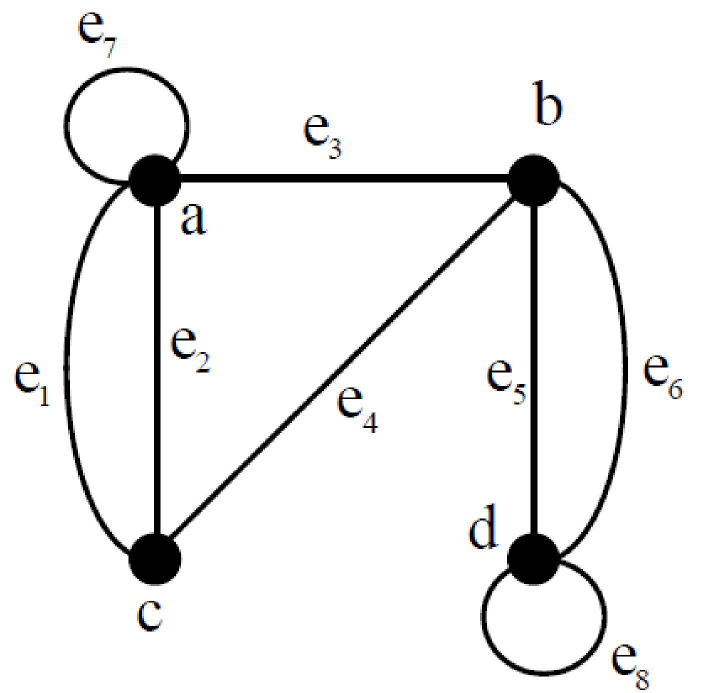
Há ainda os multigrafos (nos quais múltiplas arestas paralelas são permitidas)





Observação

Ou ainda os pseudografos (não orientados nos quais *self-loops* são permitidos)

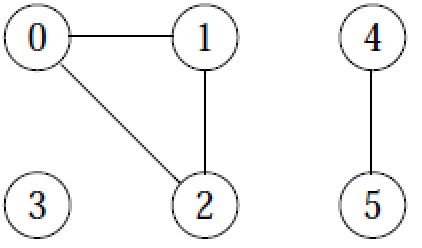


Observação

Nesta disciplina iremos estudar apenas os grafos simples (a menos que algo diferente seja explitamente mencionado, por exemplo em exercícios ou EP's)

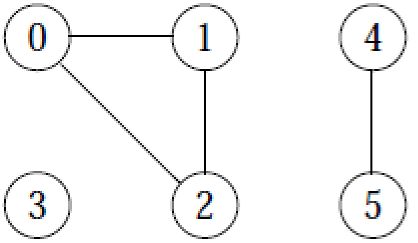
Grau de um Vértice

- Em grafos não direcionados:
  - O grau de um vértice é o número de arestas que incidem nele.
  - Um vérice de grau zero é dito **isolado** ou **não conectado**.

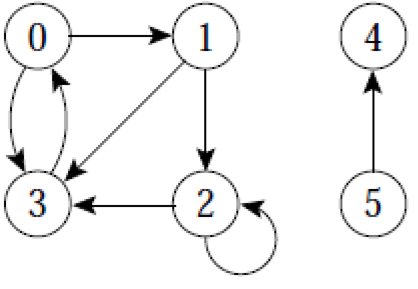


Grau de um Vértice

- Em grafos não direcionados:
  - O grau de um vértice é o número de arestas que incidem nele.
  - Um vérice de grau zero é dito **isolado** ou **não conectado**.
  - Ex.: O vértice 1 tem grau 2 e o vértice 3 é isolado.

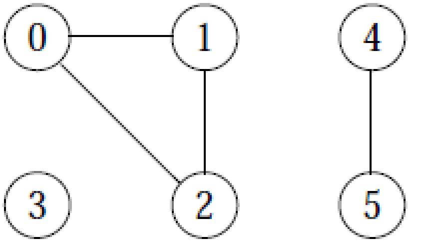


- Em grafos direcionados
  - O grau de um vértice é o número de arestas que saem dele (*out-degree*) mais o número de arestas que chegam nele (*in-degree*).
  - Ex.: O vértice 2 tem *in-degree*

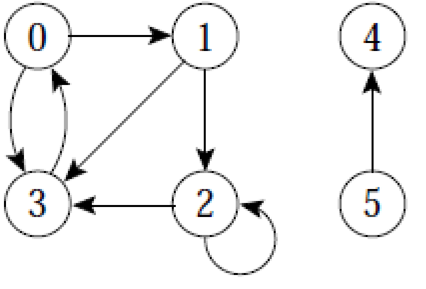


Grau de um Vértice

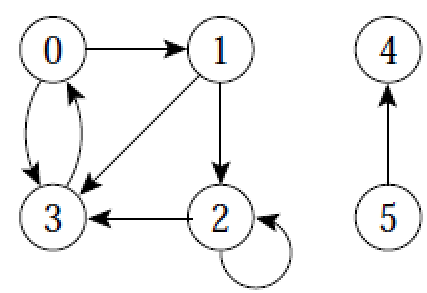
- Em grafos não direcionados:
  - O grau de um vértice é o número de arestas que incidem nele.
  - Um vérice de grau zero é dito **isolado** ou **não conectado**.
  - Ex.: O vértice 1 tem grau 2 e o vértice 3 é isolado.



- Em grafos direcionados
  - O grau de um vértice é o número de arestas que saem dele (*out-degree*) mais o número de arestas que chegam nele (*in-degree*).
  - Ex.: O vértice 2 tem *in-degree* 2, *out-degree* 2 e grau 4.



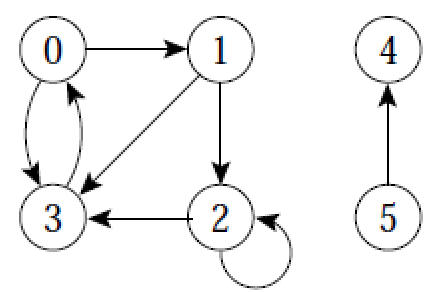
Caminho entre Vértices



Caminho entre Vértices

- Um caminho de **comprimento** k de um vértice x a um vértice y em um grafo $G = (V, A)$ é

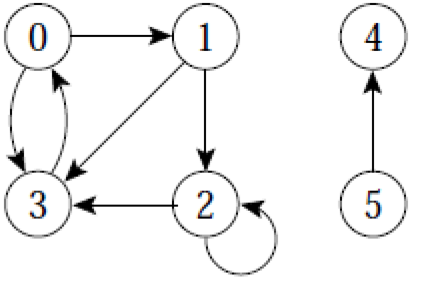
- O comprimento de um caminho é o número de arestas nele,



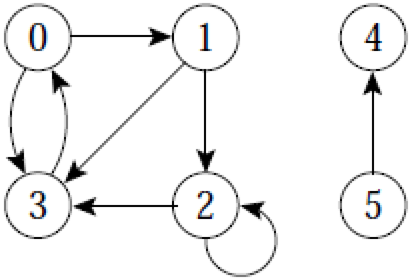
Caminho entre Vértices

- Um caminho de **comprimento** k de um vértice x a um vértice y em um grafo $G = (V, A)$ é uma sequência de vértices $(v_0, v_1, v_2, \ldots, v_k)$ tal que $x = v_0$ e $y = v_k$, e $(v_{i-1}, v_i) \in A$ para $i = 1, 2, \ldots, k$.
- O comprimento de um caminho é o número de arestas nele, isto é, o caminho contém os vértices $v_0, v_1, v_2, \ldots, v_k$ e as arestas $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$.
- Se existir um caminho c de x a y então y é **alcançável** a partir de x via c.
- Um caminho é **simples** se todos os vértices do caminho são distintos.

Ex.: O caminho $(0, 1, 2, 3)$ é simples e tem comprimento 3. O caminho $(1, 3, 0, 3)$ não é simples.

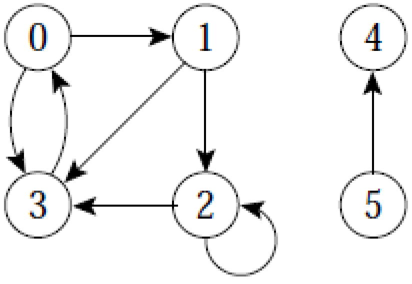


Ciclos



Ciclos

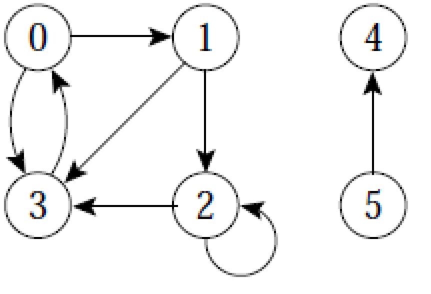
- Em um grafo direcionado:
  - Um caminho $(v_0, v_1, \ldots, v_k)$ forma um ciclo se $v_0 = v_k$ e o caminho contém pelo menos uma aresta.
  - O ciclo é simples se os vértices $v_1, v_2, \ldots, v_k$ são distintos.
  - O *self-loop* é um ciclo de tamanho 1.
  - Dois caminhos $(v_0, v_1, \ldots, v_k)$ e $(v'_0, v'_1, \ldots, v'_k)$ formam o mesmo ciclo se



Ciclos

- Em um grafo direcionado:
  - Um caminho $(v_0, v_1, \ldots, v_k)$ forma um ciclo se $v_0 = v_k$ e o caminho contém pelo menos uma aresta.
  - O ciclo é simples se os vértices $v_1, v_2, \ldots, v_k$ são distintos.
  - O *self-loop* é um ciclo de tamanho 1.
  - Dois caminhos $(v_0, v_1, \ldots, v_k)$ e $(v'_0, v'_1, \ldots, v'_k)$ formam o mesmo ciclo se existir um inteiro j tal que $v'_i = v_{(i+j) \bmod k}$ para $i = 0, 1, \ldots, k-1$.

Ex.: O caminho $(0, 1, 2, 3, 0)$ forma um ciclo. O caminho$(0, 1, 3, 0)$ forma o mesmo ciclo que os caminhos $(1, 3, 0, 1)$ e $(3, 0, 1, 3)$.



Referências

Livro do Ziviani (cap 7)