ACH2024 –

Algoritmos e Estruturas de Dados II

Prof. Helton Hideraldo Bíscaro

Objetivo e Programa

- Objetivo:

  Estudo e resolução de problemas que utilizem estruturas de dados complexas. Desenvolvimento e implementação de algoritmos clássicos.

- Programa:

  Algoritmos para classificação externa em disco e fita. Arquivos, consultas, organizações sequênciais, técnicas de indexação, indexação cilindro-superfície, árvores-B, tries e hashing. Organização de arquivos: sequêncial, aleatória e invertida. Estruturas de dados para alocação dinâmica de memória, coleta e compactação de lixo. Estruturas de dados para representação de grafos, algoritmos de busca em grafos.

- Programação em C

Avaliação

- 2 provas:

  MP = (P1 + P2)/2 + média de algumas provas surpresa

- 2 Exercícios programas (EPs) individuais:

  MT = (EP1 + EP2)/2

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

  Se MP >= 5 E MT >= 5

  M1 = (7*MP + 3* MT)/10

  senão M1 = min(4.0, [7*MP + 3* MT]/10)

- Prova SUB:

  - só para quem perdeu alguma prova (não precisa de atestado, mas prefira não fazê-la, é mais difícil que as outras)

- RECuperação (Média 2ª avaliação M2):

  M2 = (M1 + REC)/2

Participação

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

Recursos

- TIDIA:
  - PDFs das aula no “Repositório” (mas façam anotações em seus cadernos!)
  - Entrega de EPs em “Atividades”
  - Emails em “Mensagens” (usar sempre “com cópia para o e-mail do destinatário)

Bibliografia

- Bibliografia descrita na ementa da disciplina no Jupiter, incluindo:

Bibliografia:

- ZIVIANI, N. Projetos de Algoritmos - com implementações em Java e C++. Thomson Learning, 2007.
- 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.

Parte 1 da Disciplina:

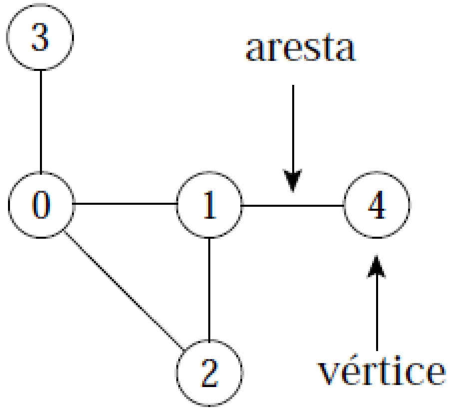
GRAFOS

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.



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

Grafos

- Então para que pode servir isso?

Grafos - Motivação

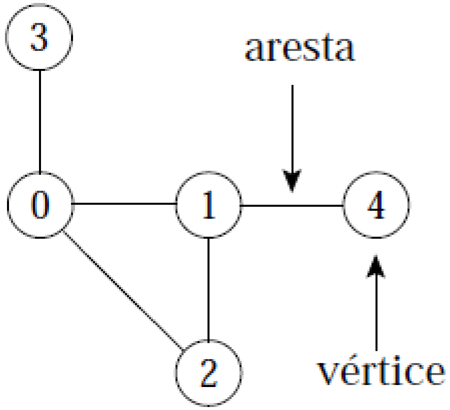
- Muitas aplicações em computação necessitam considerar conjunto de conexões entre pares de objetos:
  - Existe um caminho para ir de um objeto a outro seguindo as conexões?
  - Qual é a menor distância entre um objeto e outro objeto?
  - Quantos outros objetos podem ser alcançados a partir de um determinado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

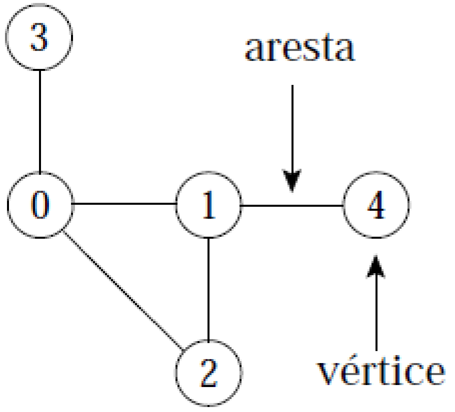
- **Grafo**: conjunto de vértices e arestas.
- **Vértice**: objeto simples que pode ter nome e outros atributos.
- **Aresta**: conexão entre dois vértices.



- 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.

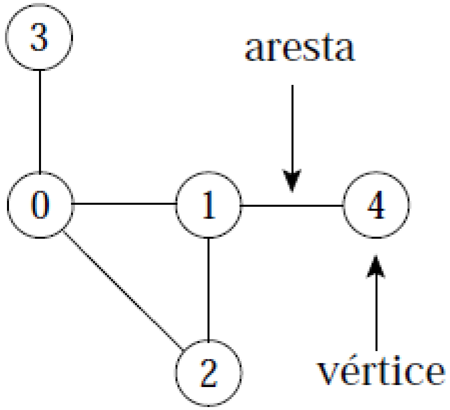


- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

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.



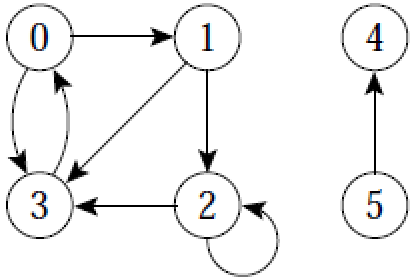
- Notação: $G = (V, A)$
  - G: grafo
  - V: conjunto de vértices
  - A: conjunto de arestas

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….

Grafos Direcionados

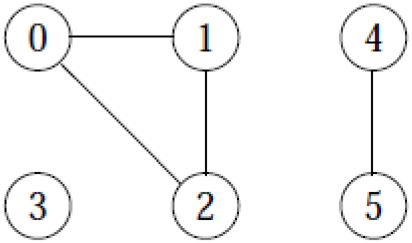
- Um **grafo direcionado** G é um par (V, A), onde V é um conjunto finito de vértices e A é uma relação binária em V.
  - Uma aresta (u, v) sai do vértice u e entra no vértice v. O vértice v é **adjacente** ao vértice u.
  - Podem existir arestas de um vértice para ele mesmo, chamadas de *self-loops*.



Também chamados **Digrafos**

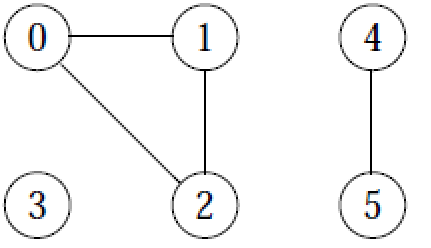
Grafos Não Direcionados

- 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.

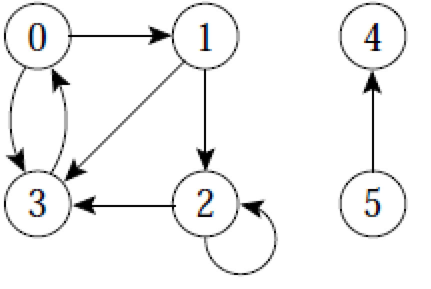


Grau de um Vértice

- 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

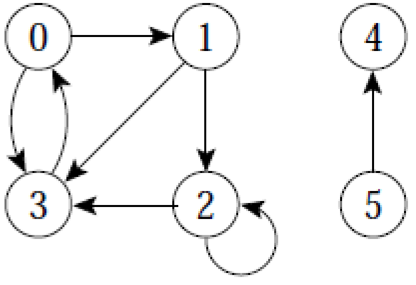
- Um caminho de **comprimento** k de um vértice x a um vértice y em um grafo $G = (V, A)$ é uma sequência de vértices $(v_0, v_1, v_2, \ldots, v_k)$ tal que $x = v_0$ e $y = v_k$, e $(v_{i-1}, v_i) \in A$ para $i = 1, 2, \ldots, k$.

- 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.

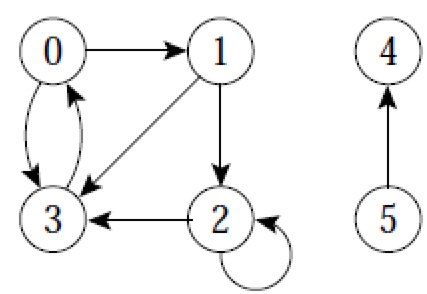


Ciclos

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 existir um inteiro j tal que $v'_i = v_{(i+j) \bmod k}$ para $i = 0, 1, \ldots, k-1$.

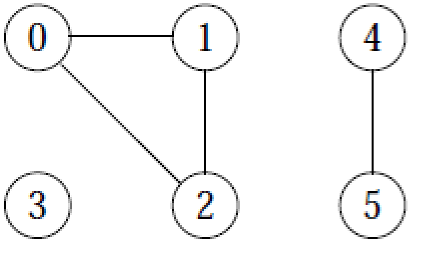
Ex.: O caminho $(0, 1, 2, 3, 0)$ forma um ciclo.
O caminho$(0, 1, 3, 0)$ forma o mesmo ciclo que os caminhos $(1, 3, 0, 1)$ e $(3, 0, 1, 3)$.



Ciclos

- Em um grafo não direcionado:
  - Um caminho $(v_0, v_1, \ldots, v_k)$ forma um ciclo se $v_0 = v_k$ e o caminho contém pelo menos três arestas.
  - O ciclo é simples se os vértices $v_1, v_2, \ldots, v_k$ são distintos.

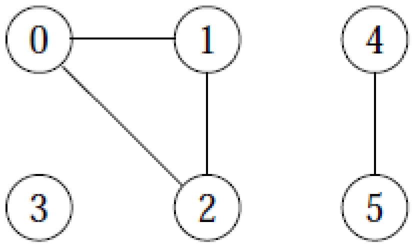
Ex.: O caminho $(0, 1, 2, 0)$ é um ciclo.



Componentes Conectados

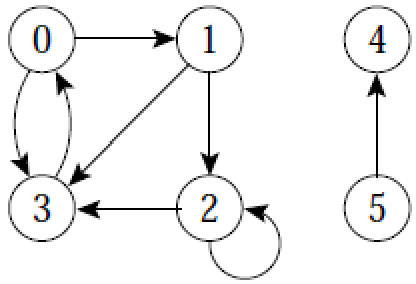
- Um grafo não direcionado é conectado se cada par de vértices está conectado por um caminho.
- Os componentes conectados são as porções conectadas de um grafo.
- Um grafo não direcionado é conectado se ele tem exatamente um componente conectado.

Ex.: Os componentes são: $\{0, 1, 2\}$, $\{4, 5\}$ e $\{3\}$.



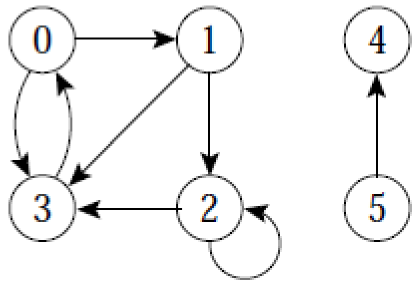
Componentes Fortemente Conectados

- Um grafo direcionado $G = (V, A)$ é **fortemente conectado** se cada dois vértices quaisquer são alcançáveis a partir um do outro.
- Os **componentes fortemente conectados** de um grafo direcionado são conjuntos de vértices sob a relação "são mutuamente alcançáveis".
- Um **grafo direcionado fortemente conectado** tem apenas um componente fortemente conectado.



Componentes Fortemente Conectados

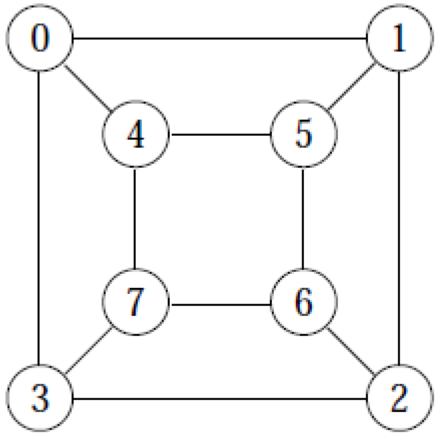
- Um grafo direcionado $G = (V, A)$ é **fortemente conectado** se cada dois vértices quaisquer são alcançáveis a partir um do outro.
- Os **componentes fortemente conectados** de um grafo direcionado são conjuntos de vértices sob a relação "são mutuamente alcançáveis".
- Um **grafo direcionado fortemente conectado** tem apenas um componente fortemente conectado.

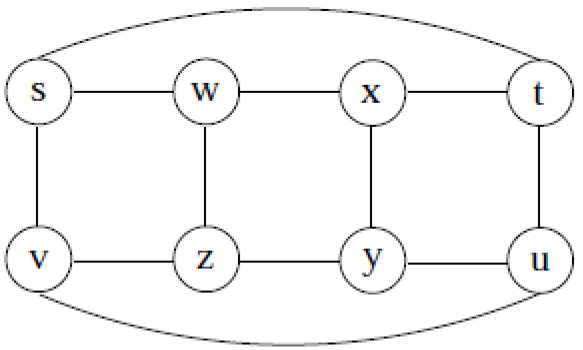


Ex.: $\{0, 1, 2, 3\}$, $\{4\}$ e $\{5\}$ são os componentes fortemente conectados, $\{4, 5\}$ não o é pois o vértice 5 não é alcançável a partir do vértice 4.

Grafos Isomorfos

- $G = (V, A)$ e $G' = (V', A')$ são isomorfos se existir uma bijeção $f : V \to V'$ tal que $(u, v) \in A$ se e somente se $(f(u), f(v)) \in A'$.
- Em outras palavras, é possível re-rotular os vértices de G para serem rótulos de G' mantendo as arestas correspondentes em G e G'.





Subgrafos

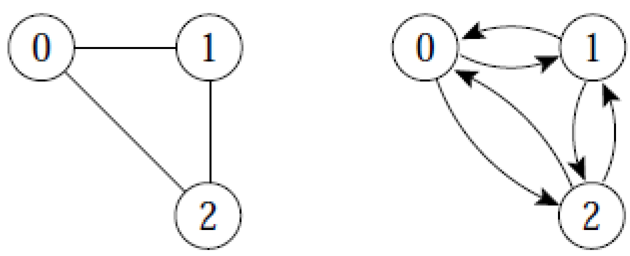
- Um grafo $G' = (V', A')$ é um subgrafo de $G = (V, A)$ se $V' \subseteq V$ e $A' \subseteq A$.
- Dado um conjunto $V' \subseteq V$, o subgrafo induzido por V' é o grafo $G' = (V', A')$, onde $A' = \{(u, v) \in A | u, v \in V'\}$.

Ex.: Subgrafo induzido pelo conjunto de vértices $\{1, 2, 4, 5\}$.

Versão Direcionada de um Grafo Não Direcionado

- A versão direcionada de um grafo não direcionado $G = (V, A)$ é um grafo direcionado $G' = (V', A')$ onde $(u, v) \in A'$ se e somente se $(u, v) \in A$.

- Cada aresta não direcionada (u, v) em G é substituída por duas arestas direcionadas (u, v) e (v, u)

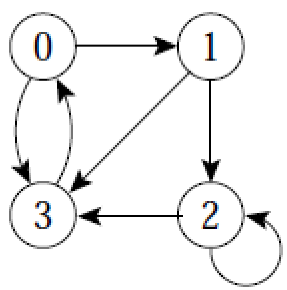


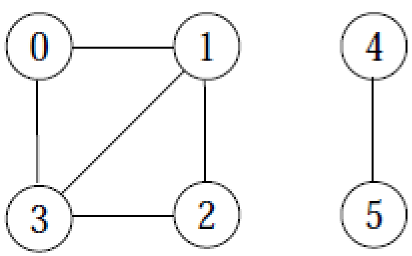
Versão Não Direcionada de um Grafo Direcionado

- A versão não direcionada de um grafo direcionado $G = (V, A)$ é um grafo não direcionado $G' = (V', A')$ onde $(u, v) \in A'$ se e somente se $u \neq v$ e $(u, v) \in A$.
- A versão não direcionada contém as arestas de G sem a direção e sem os *self-loops*.

Vizinhos e adjacentes

- Em um grafo direcionado, um **vizinho** de um vértice u é qualquer vértice adjacente a u na versão não direcionada de G.
- Em um grafo não direcionado, u e v são vizinhos se eles são adjacentes.



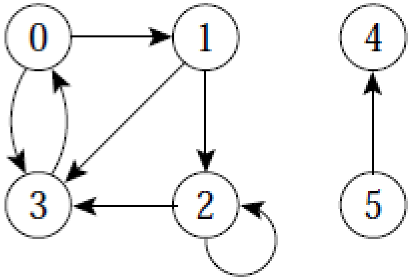


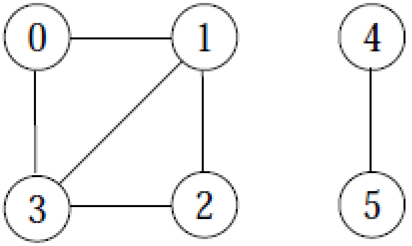
1 é adjacente a 3 ?
1 é vizinho de 3 ?
3 é adjacente a 1 ?
3 é vizinho de 1 ?

1 é adjacente a 3 ?
1 é vizinho de 3 ?
3 é adjacente a 1 ?
3 é vizinho de 1 ?

Vizinhos e adjacentes

- Em um grafo direcionado, um **vizinho** de um vértice u é qualquer vértice adjacente a u na versão não direcionada de G.
- Em um grafo não direcionado, u e v são vizinhos se eles são adjacentes.



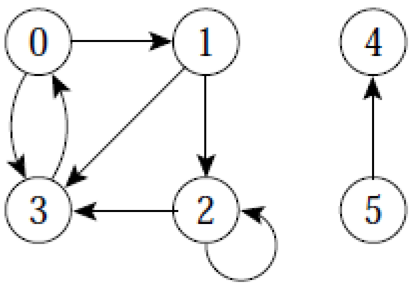


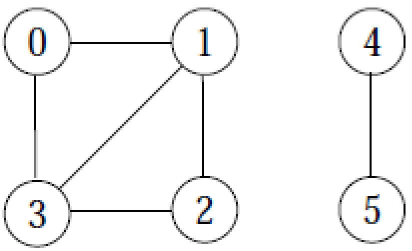
1 é adjacente a 3 ? Não
1 é vizinho de 3 ? Sim
3 é adjacente a 1 ? Sim
3 é vizinho de 1 ? Sim

1 é adjacente a 3 ? Não
1 é vizinho de 3 ? Sim
3 é adjacente a 1 ? Sim
3 é vizinho de 1 ? Sim

Vizinhos e adjacentes

- Em um grafo direcionado, um **vizinho** de um vértice u é qualquer vértice adjacente a u na versão não direcionada de G.
- Em um grafo não direcionado, u e v são vizinhos se eles são adjacentes.





1 é adjacente a 3 ? Não
1 é vizinho de 3 ? Sim
3 é adjacente a 1 ? Sim
3 é vizinho de 1 ? Sim

1 é adjacente a 3 ? Não
1 é vizinho de 3 ? Sim
3 é adjacente a 1 ? Sim
3 é vizinho de 1 ? Sim

1 e 4 não são adjacentes nem vizinhos

Grafos Completos

- Um grafo completo é um grafo não direcionado no qual todos os pares de vértices são adjacentes.
- Possui $(|V|^2 - |V|)/2 = |V|(|V| - 1)/2$ arestas

POR QUÊ?

Grafos Completos

- Um grafo completo é um grafo não direcionado no qual todos os pares de vértices são adjacentes.
- Possui $(|V|^2 - |V|)/2 = |V|(|V| - 1)/2$ arestas, pois do total de $|V|^2$ pares possíveis de vértices devemos subtrair $|V|$ *self-loops* e dividir por 2 (cada aresta ligando dois vértices é contada duas vezes).
- O número total de **grafos diferentes** com $|V|$ vértices é $2^{|V|(|V|-1)/2}$ (número de maneiras diferentes de escolher um subconjunto a partir de $|V|(|V| - 1)/2$ possíveis arestas).

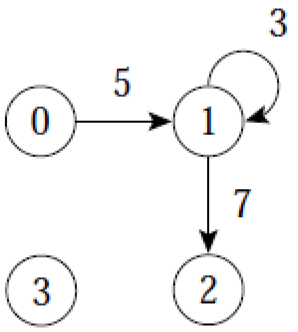
Grafos Completos

- Um grafo completo é um grafo não direcionado no qual todos os pares de vértices são adjacentes.
- Possui $(|V|^2 - |V|)/2 = |V|(|V|-1)/2$ arestas, pois do total de $|V|^2$ pares possíveis de vértices devemos subtrair $|V|$ *self-loops* e dividir por 2 (cada aresta ligando dois vértices é contada duas vezes).
- O número total de **grafos diferentes** com $|V|$ vértices é $2^{|V|(|V|-1)/2}$ (número de maneiras diferentes de escolher um subconjunto a partir de $|V|(|V|-1)/2$ possíveis arestas).

Faria sentido falarmos em grafos completos direcionados?

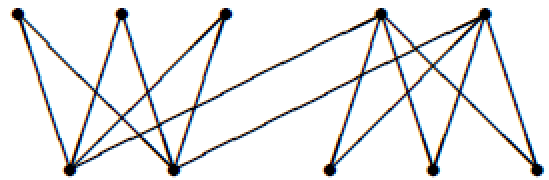
Grafo ponderado

possui pesos associados às arestas.



Grafo Bipartido

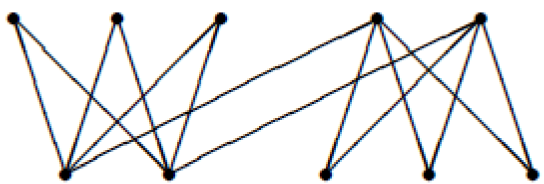
- **Grafo bipartido**: grafo não direcionado $G = (V, A)$ no qual V pode ser particionado em dois conjuntos V_1 e V_2 tal que $(u, v) \in A$ implica que $u \in V_1$ e $v \in V_2$ ou $u \in V_2$ e $v \in V_1$ (todas as arestas ligam os dois conjuntos V_1 e V_2).



bipartido

Grafo Bipartido

- **Grafo bipartido**: grafo não direcionado $G = (V, A)$ no qual V pode ser particionado em dois conjuntos V_1 e V_2 tal que $(u, v) \in A$ implica que $u \in V_1$ e $v \in V_2$ ou $u \in V_2$ e $v \in V_1$ (todas as arestas ligam os dois conjuntos V_1 e V_2).



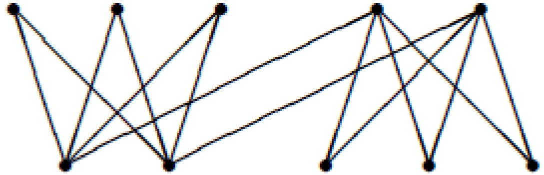
bipartido

?

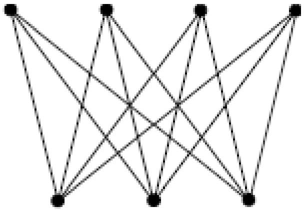
bipartido completo

Grafo Bipartido

- **Grafo bipartido**: grafo não direcionado $G = (V, A)$ no qual V pode ser particionado em dois conjuntos V_1 e V_2 tal que $(u, v) \in A$ implica que $u \in V_1$ e $v \in V_2$ ou $u \in V_2$ e $v \in V_1$ (todas as arestas ligam os dois conjuntos V_1 e V_2).



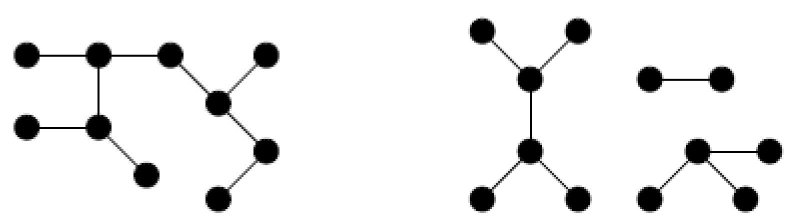
bipartido



bipartido completo

Árvores

- **Árvore livre**: grafo não direcionado acíclico e conectado. É comum dizer apenas que o grafo é uma árvore omitindo o "livre".
- **Floresta**: grafo não direcionado acíclico, podendo ou não ser conectado.
- **Árvore geradora** de um grafo conectado $G = (V, A)$: subgrafo que contém todos os vértices de G e forma uma árvore.
- **Floresta geradora** de um grafo $G = (V, A)$: subgrafo que contém todos os vértices de G e forma uma floresta.



(a) (b)