Universidade Federal do Vale do São Francisco
Curso de Engenharia da Computação

Matemática Discreta – 10

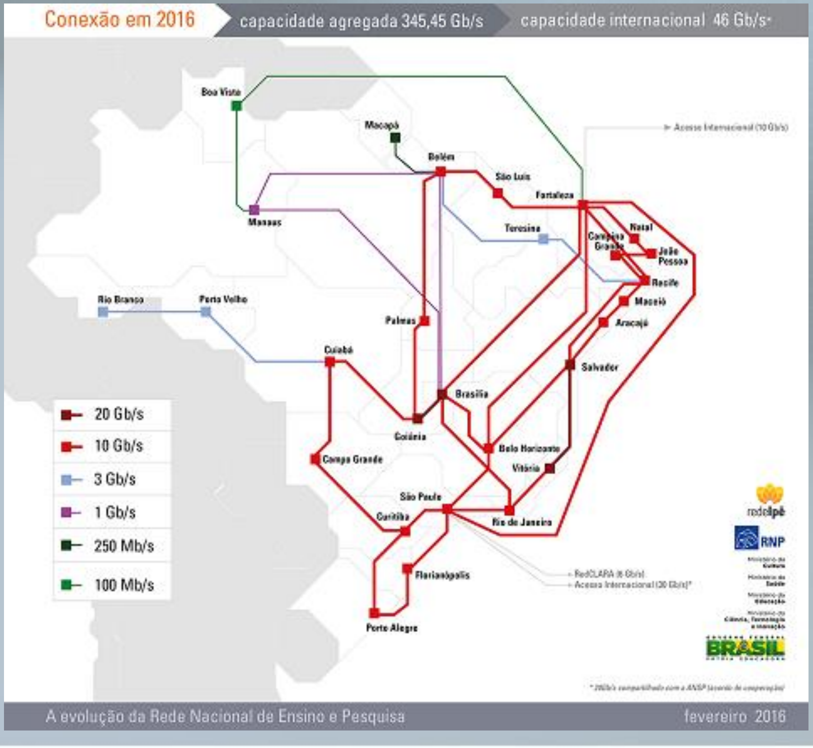
Prof. Jorge Cavalcanti
jorge.cavalcanti@univasf.edu.br - www.univasf.edu.br/~jorge.cavalcanti

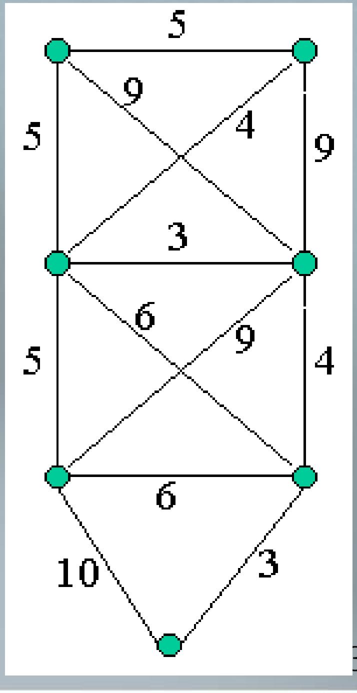
Grafos e Árvores

- 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?
  - Quantos outros objetos podem ser alcançados a partir de um determinado objeto?
- Grafos são utilizados para modelar tais problemas.
- 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 qual é o roteiro mais curto para visitar as principais cidades de uma região turística.

Grafos e Árvores

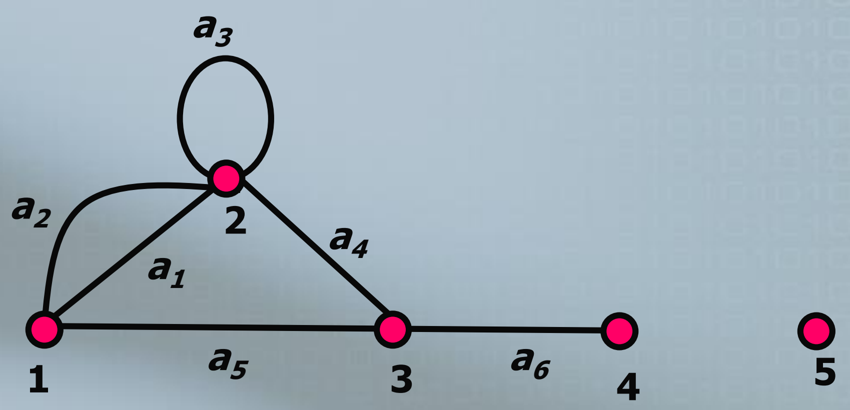
- **Definição informal** - Um grafo é um conjunto não-vazio de nós (vértices) e um conjunto de arcos (arestas) tais que cada arco conecta dois nós.
  - Os grafos que serão estudados terão sempre um número finito de nós e arcos.





Grafos e Árvores

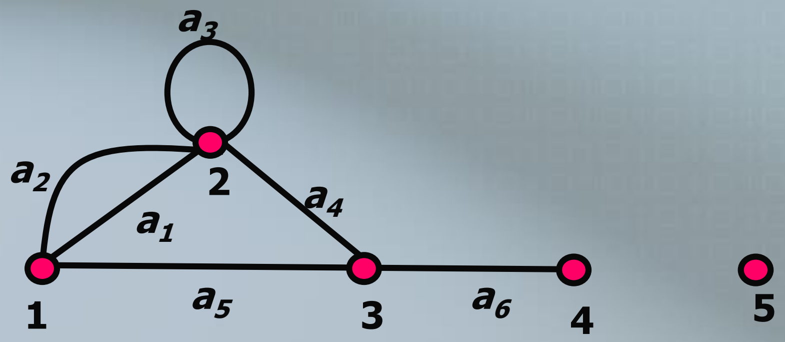
- O grafo a seguir tem cinco nós e seis arcos:



- A definição informal de um grafo funciona bem se tivermos sua representação visual, mostrando que arcos se conectam aos nós.
- Sem essa visualização, precisamos de uma definição formal de mostrar esse grafo.

Grafos e Árvores

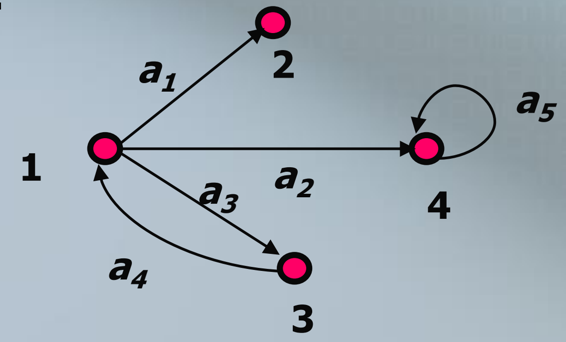
- **Definição Formal** - Um grafo é uma tripla ordenada (N, A, g), onde:
  - N = um conjunto não-vazio de nós (**vértices**)
  - A = um conjunto de arcos (**arestas**)
  - g = uma função que associa a cada arco ***a*** a um par **não-ordenado** x-y de nós, chamado de **extremidades** de a



- Ex. 01: No grafo acima, a função ***g*** que associa arcos a suas extremidades é a seguinte: $g(a_1)$=1-2, $g(a_2)$=1-2, $g(a_3)$=2-2, $g(a_4)$=2-3, $g(a_5)$=1-3 *e* $g(a_6)$=3-4.

Grafos e Árvores

- **Grafo direcionado (dígrafo) –** Um grafo é uma tripla ordenada (N, A, g), onde
  - N = um conjunto não-vazio de nós (**vértices**)
  - A = um conjunto de arcos (**arestas**)
  - *g* = uma função que associa a cada arco ***a*** a um **par ordenado** (x, y) de nós, onde **x** é o ponto inicial e **y** é o ponto final de ***a***.
  - Em um grafo direcionado, cada arco tem um sentido ou orientação.

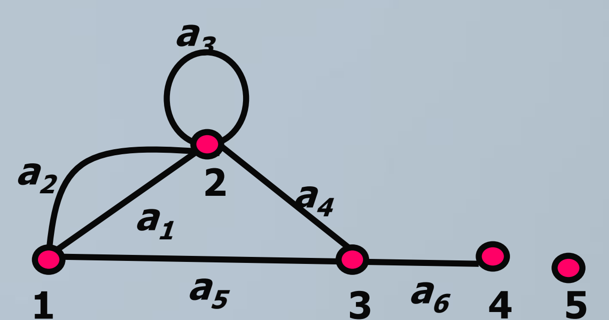


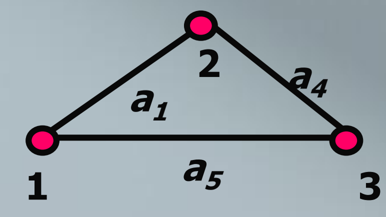
- Ex. 02: No grafo acima, a função ***g*** que associa arcos a suas extremidades é a seguinte: $g(a_1)=(1,2)$, $g(a_2)=(1,4)$, $g(a_3)=(1,3)$, $g(a_4)=(3,1)$ e $g(a_5)=(4,4)$.

Grafos e Árvores

Terminologia

- Além da orientação, podemos colocar informações nos nós (rótulos), gerando um **grafo rotulado**. Pode-se também atribuir valores ou pesos aos arcos, gerando um **grafo com pesos**.
- **Nós adjacentes** – se ambos são extremidades de algum arco.
- **Laço** - é um arco com extremidades *n-n* para algum nó *n*.
- **Arcos paralelos** – dois arcos com a mesma extremidade.
- **Grafo Simples** – é um grafo sem laços ou arcos paralelos.
- **Nó isolado –** é um nó que não é adjacente a nenhum outro.
- **Grau** – é o número de extremidades de arcos que se conectam a um nó.
- **Grafo completo -** é um grafo no qual dois nós distintos quaisquer são adjacentes.
- **Subgrafo –** consiste em um conjunto de nós e arcos que são subconjuntos do conjunto original de nós e arcos.





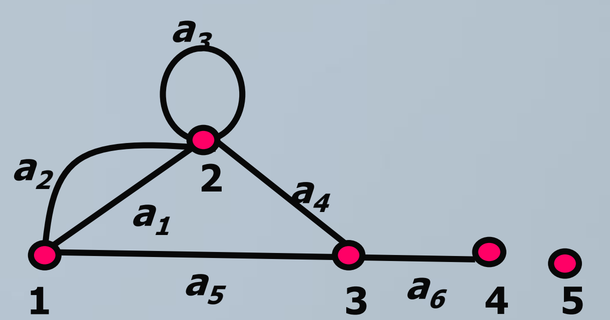
Grafos e Árvores

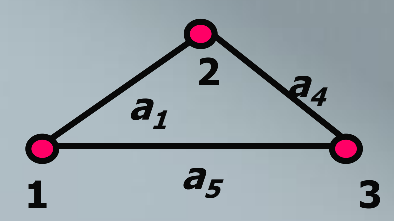
Terminologia

- **Caminho** – do nó n_0 para o nó n_k é uma sequência:

  $n_0, a_0, n_1, a_1, .. N_{k-1}, a_{k-1}, n_k$

  - O **comprimento** de um caminho é o número de arcos que ele contém.
- **Grafo conexo** – se existe um caminho de qualquer nó para outro.
- **Ciclo** – é um caminho de algum nó n_0 para ele mesmo tal que nenhum arco aparece mais de uma vez.
  - n_0 é o único nó que aparece mais de uma vez e apenas nas extremidades.
  - Um grafo sem ciclos é dito **acíclico.**





Grafos e Árvores

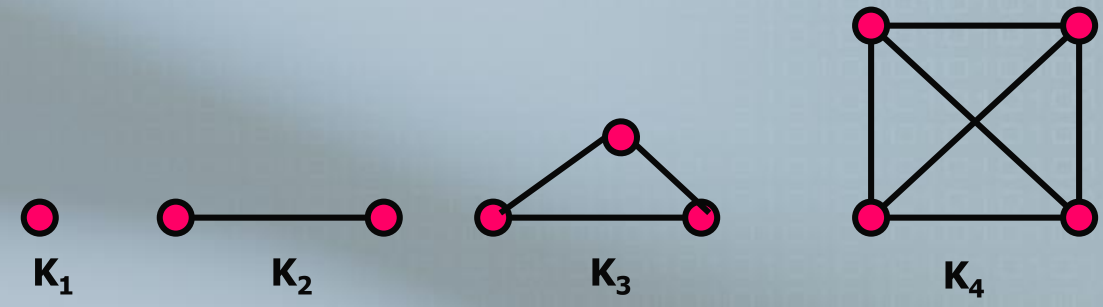
Exercício

- Ex. 03: Esboce um grafo com nós {1,2,3,4,5}, arcos {a_1, a_2, a_3, a_4, a_5, a_6} e função *g*, dada por $g(a_1)$=1-2, $g(a_2)$=1-3, $g(a_3)$=3-4, $g(a_4)$=3-4, $g(a_5)$=4-5 *e* $g(a_6)$=5-5. Depois responda o que se segue:
  a) Encontre 2 nós que não são adjacentes
  b) Encontre um nó adjacente a si mesmo
  c) Encontre um laço
  d) Encontre 2 arcos paralelos
  e) Encontre o nó de grau 3
  f) Encontre um caminho de comprimento 5
  g) Encontre um ciclo
  h) Esse grafo é completo?
  i) Esse grafo é conexo?

Grafos e Árvores

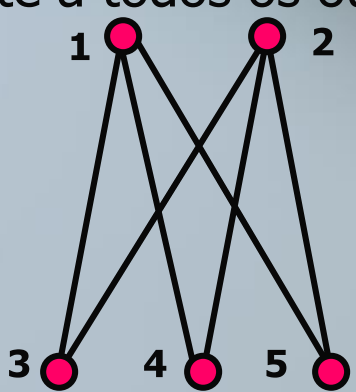
Terminologia

- As figuras abaixo ilustram os grafos simples completos de 1 a 4 vértices. Um grafo simples completo é denotado por K_n.



K_5 = ?

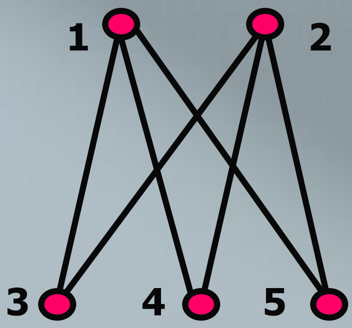
- O grafo simples da figura abaixo não é completo, pois nem todo nó é adjacente a todos os outros.



Grafos e Árvores

Terminologia

- Entretanto, os nós podem ser divididos em 2 conjuntos disjuntos {1,2} e {3,4,5}, tais que os nós de cada conjunto não são adjacentes, mas dois nós escolhidos um em cada conjunto são adjacentes.
  - Esse tipo de grafo é chamado de bipartido completo.
- **Grafo bipartido completo** – se os seus nós podem ser divididos em 2 conjuntos disjuntos não-vazios N_1 (*m elementos*) e N_2 (*n elementos*), tais que 2 nós são adjacentes se, e somente se, um deles pertence a N_1 e o outro pertence a N_2.
  - Um tal grafo é denotado por $K_{m,n}$

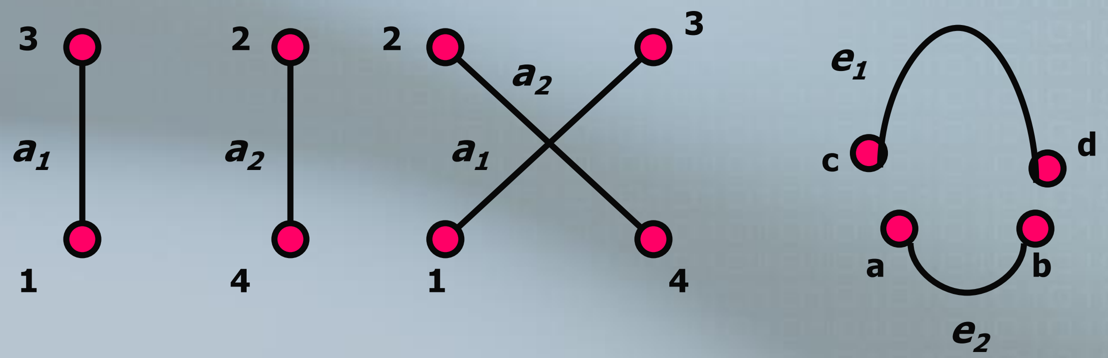


$K_{2,3}$

Grafos e Árvores

Terminologia

- Dois grafos podem parecer diferentes na sua representação visual, mas podem ser o mesmo grafo de acordo com sua representação formal.



f_1: 1 → a
2 → c
3 → b
4 → d

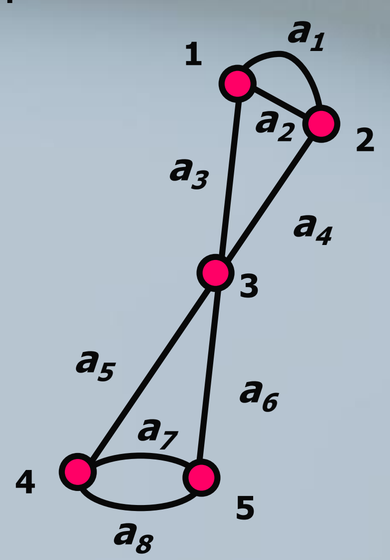
f_2: a1 → e2
a2 → e1

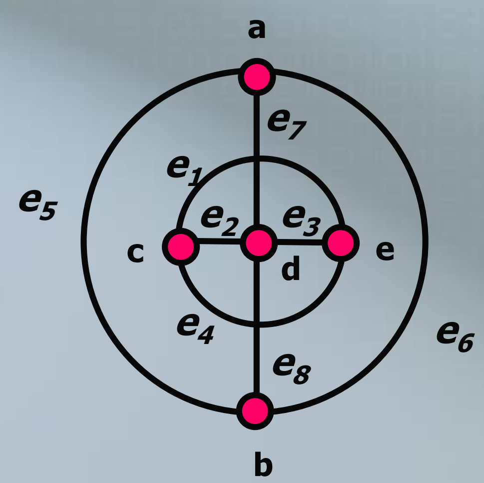
- **Grafos Isomorfos** – dois grafos (N_1, A_1, g_1) e (N_2, A_2, g_2) são isomorfos se existem bijeções $f_1\!:N_1 \rightarrow N_2$ e $f_2\!: A_1 \rightarrow A_2$ tais que, para cada arco $a \in A_1$, $g_1(a) = x\text{-}y$ se, e somente se $g_2[f_2(a)] = f_1(x)\text{-}f_1(y)$.

Grafos e Árvores

Terminologia

- Em outras palavras, deve ser possível re-rotular os nós de um grafo para serem rótulos de outro, mantendo os arcos correspondentes em cada grafo.
- Ex. 04: Nos grafos isomorfos abaixo, complete as bijeções que estabelecem o isomorfismo.





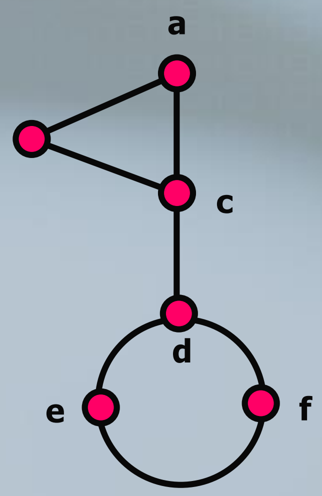
f_1: 1 → c
2 → e
3 → d
4 → b
5 → a

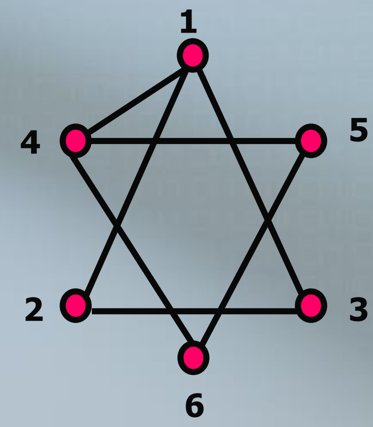
f_2: a1 → e1
a2 → e4
⋮

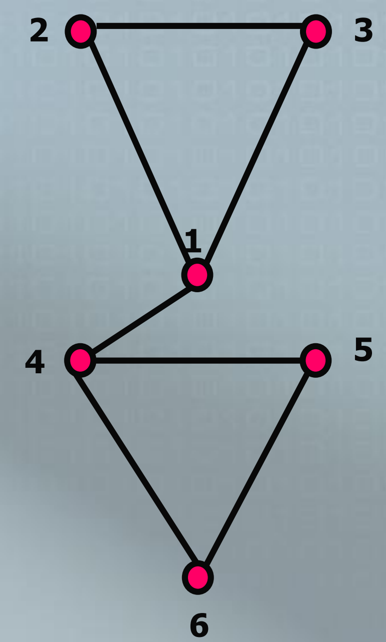
Grafos e Árvores

Terminologia

- Ex. 05: Nos grafos abaixo verifique se são isomorfos e, em caso positivo, descreva as bijeções que estabelecem o isomorfismo.







Grafos e Árvores

Terminologia

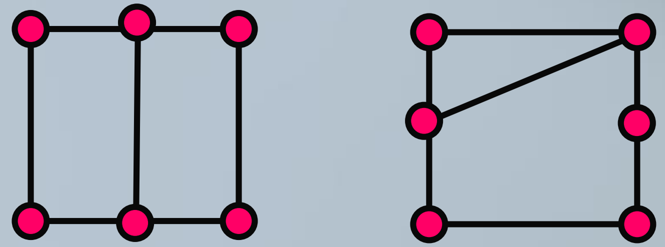
- **Teorema sobre Isomorfismo de Grafos Simples**
  - Dois grafos simples (N_1, A_1, g_1) e (N_2, A_2, g_2) são isomorfos se existem bijeções $f_1{:}N_1 \rightarrow N_2$ tal que, quaisquer que sejam os nós n_i e n_j de N_1, n_i e n_j são adjacentes se, e somente se, $f(n_i)$ e $f(n_j)$ são adjacentes.
  - A função f é chamada de um **isomorfismo** do grafo 1 no grafo 2.
- Para provar que dois grafos são isomorfos é necessário encontrar a bijeção e depois mostrar que a propriedade de adjacência é preservada.
- Por outro lado, provar que dois grafos não são isomorfos, é preciso mostrar que as bijeções necessárias não existem. Esse método pode ser inviável em grafos maiores.
- Existem algumas condições que deixam claro que os grafos não são isomorfos, tais como:

Grafos e Árvores

Condições de não isomorfismo

1. Um grafo tem mais nós que o outro.
2. Um grafo tem mais arcos que o outro.
3. Um grafo tem arcos paralelos e o outro não.
4. Um grafo tem um laço e o outro não.
5. Um grafo tem um nó de grau *k* e o outro não.
6. Um grafo é conexo e o outro não.
7. Um grafo tem um ciclo e o outro não.

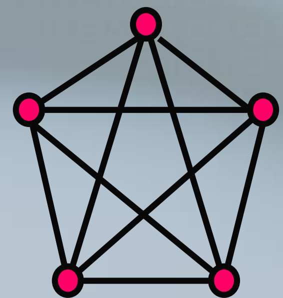
- Mesmo assim, ainda podemos falhar..

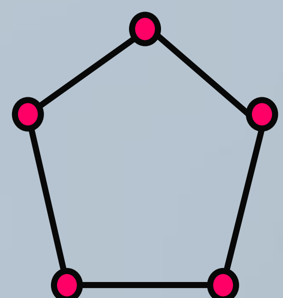


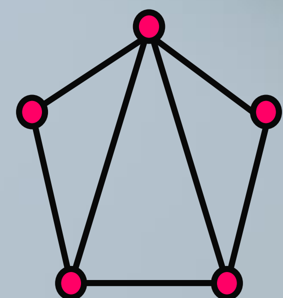
(a) (b)

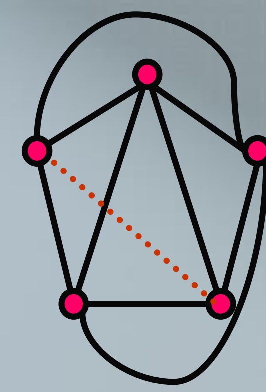
Grafos e Árvores

- **Grafo planar** – é um grafo que pode ser representado de modo que seus arcos se intersectam apenas em nós.
  - Um grafo isomorfo a um grafo planar também é planar.
- Ex. 06: Mostre que K_4 é um grafo planar.
- Ex. 07: K_5 também é planar ?









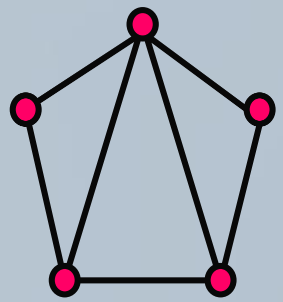
Outra Forma ?

Grafos e Árvores

- O matemático suíço Leonard Euler descobriu que um grafo planar, simples e conexo, divide o plano em um determinado número de regiões, incluindo regiões limitadas por arcos e uma região exterior ilimitada.
- Euler observou uma relação entre o número ***n*** de nós, o número ***a*** de arcos e o número ***r*** de regiões em um tal grafo.

Fórmula de Euler: n – a + r = 2

- Verifique a Fórmula de Euler no grafo abaixo:



Grafos e Árvores

Representação de grafos no computador

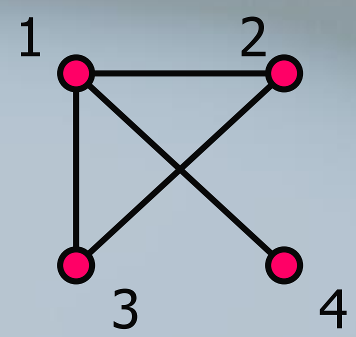
- Maior vantagem do grafo é a sua representação visual da informação.
- E se quisermos armazenar o grafo em forma digital?
  - Imagem digital – Difícil manipulação e ocupa mais espaço.
- O que precisamos é armazenar os dados essenciais que fazem parte da definição do grafo.
  - Os nós e quais são extremidades de arcos e outras informações pertinentes (pesos, cores etc.).
- As representações computacionais usuais envolvem uma das estruturas de dados:
  - Matriz de adjacência
  - Lista de adjacências

Grafos e Árvores

Representação de grafos no computador

Matriz de Adjacência

- Seja um grafo com n nós numerados ($n_1, n_2..,n_n$) arbitrariamente. Após a ordenação dos nós, podemos formar uma matriz n x n onde o elemento i, j é o número de arcos entre os nós n_i e n_j.



$$A = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \end{array}$$

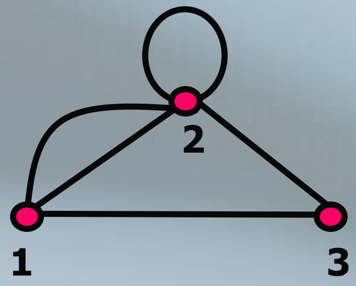
- A matriz de um grafo não-direcionado é simétrica

Grafos e Árvores

Representação de grafos no computador

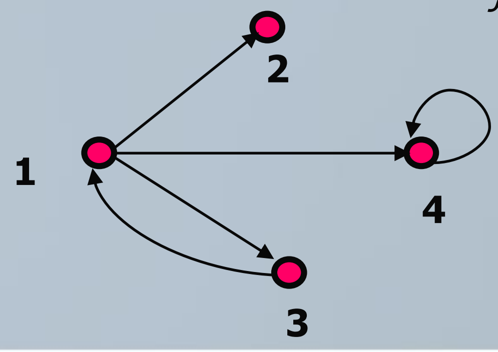
Matriz de Adjacência

- Encontre a matriz de adjacência para o grafo abaixo:



$$A = \begin{vmatrix} 0 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix}$$

- A matriz de adjacência de um grafo direcionado não será simétrica, pois a existência de um arco de n_i para n_j não implica em um arco de n_j para n_i.



$$A = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Grafos e Árvores

Representação de grafos no computador

Matriz de Adjacência

1. **Vantagens:**

- Fácil visualização para vértices adjacentes
  - Muito útil para algoritmos em que necessitamos saber com rapidez se existe uma aresta ligando dois vértices
- Fácil cálculo do grau do nó.
  - A soma dos números de uma linha retorna o grau do vértice, em grafos não direcionados
  - Em grafos direcionados
    - A soma dos números de uma linha retorna o grau de saída
    - A soma dos números de uma coluna retorna o grau de entrada

2. **Desvantagens:**

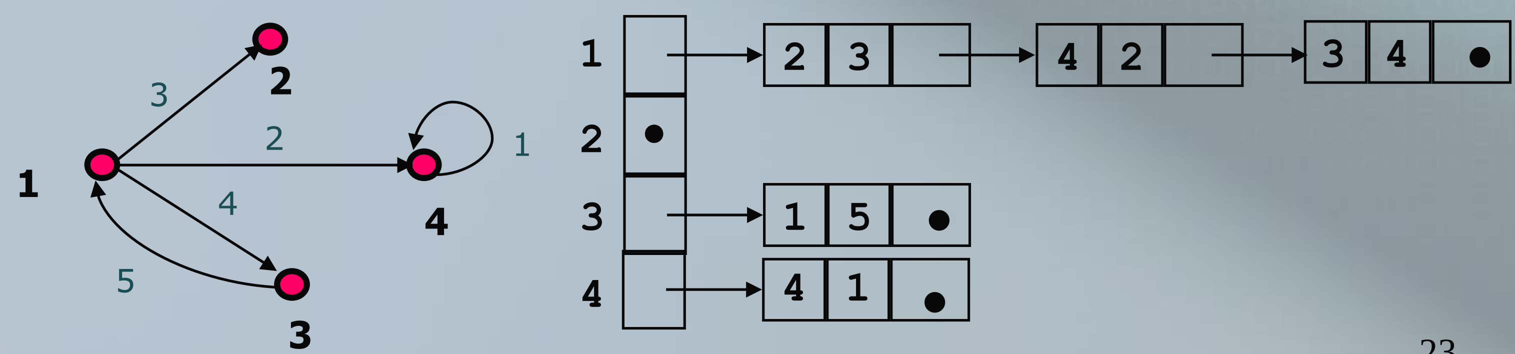
- Requer espaço de armazenamento das estruturas de dados apropriadas.
  - Deve ser mais utilizada para grafos densos

Grafos e Árvores

Representação de grafos no computador

Lista de Adjacências

- Se um grafo tem n nós, precisamos de n^2 dados para representar a matriz (ou $n^2/2$), mesmo que muitos desses dados seja igual a zero.
- Um grafo com poucos arcos pode ser representado de modo mais eficiente armazenando-se somente os elementos não nulos da matriz de adjacência.
- Essa representação consiste em uma lista, para cada nó, de todos os nós adjacentes a ele. Cada linha da matriz representa uma lista.



Grafos e Árvores

Representação de grafos no computador

Lista de Adjacências

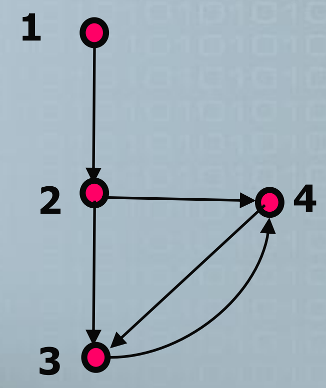
- Mais utilizada para grafos esparsos, pois também exige muito espaço para armazenamento
- Verificação de grau:
  - Não Direcionais: quantidade de nós em uma linha
  - Direcionais: A quantidade de nós de uma linha representa o grau de saída.
    - Como saber o grau de entrada de cada nó?
    - *Deve-se pesquisar em todos os vértices do grafo, excluindo ele, se existe alguma referência para o nó em questão!!!*

Grafos e Árvores

Representação de grafos no computador

Exercícios

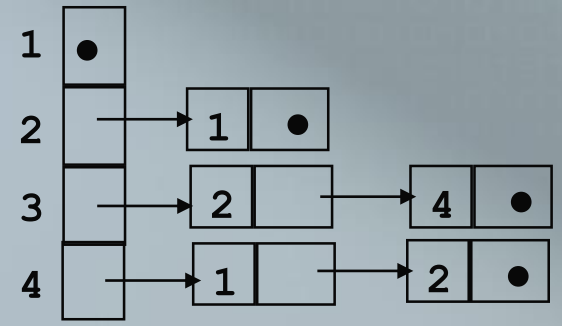
1. Escreva a matriz e a lista de adjacência do seguinte grafo:



2. Desenhe o grafo representado pela matriz de adjacência:

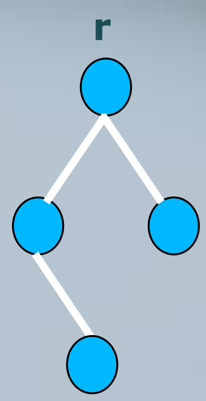
$$A = \begin{vmatrix} 0 & 2 & 0 \\ 2 & 0 & 2 \\ 0 & 2 & 0 \end{vmatrix}$$

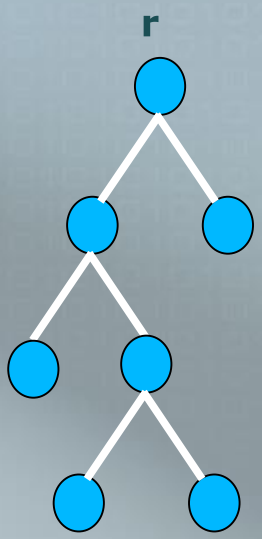
3. Desenhe o grafo direcionado representado pela lista de adjacência a seguir:



Grafos e Árvores

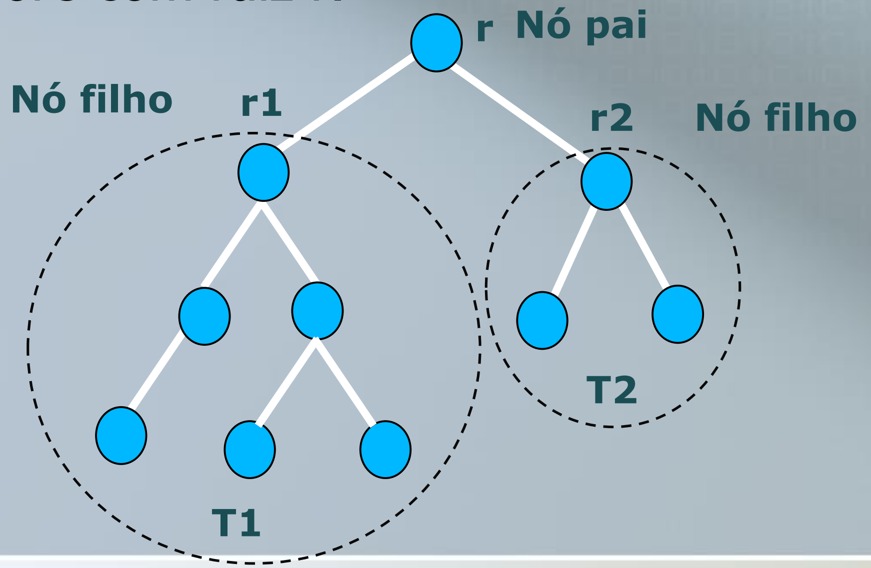
- **Árvores e suas representações**
  - Árvore é um tipo especial de grafo, útil na representação de dados
  - Por definição – é um grafo conexo, acíclico e com um nó especial, denominado de **raiz.**





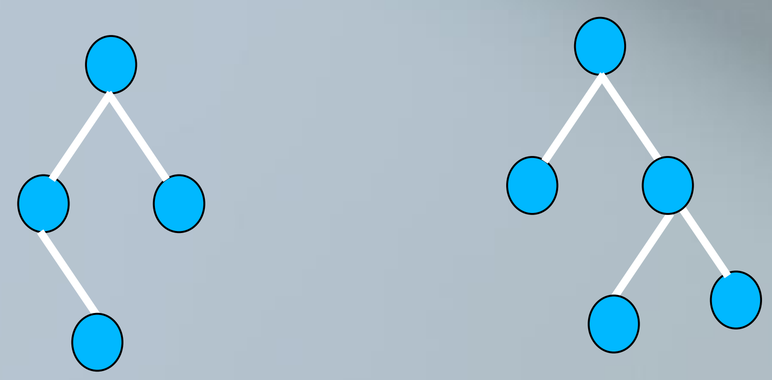
Grafos e Árvores

- **Árvores e suas representações**
  - Uma árvore também pode ser definida de maneira recorrente. O único nó é uma árvore (esse nó como raiz).
  - Sejam T1, T2, ...Tt árvores disjuntas com raízes r1, r2,... rt. Um grafo formado colocando-se um novo nó r, ligado, por um único arco a cada um dos nós r1, r2...r é uma árvore com raiz r.



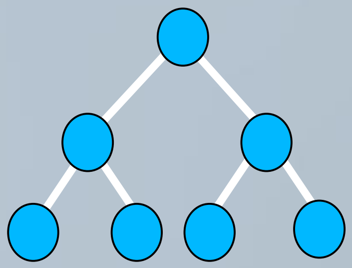
Grafos e Árvores

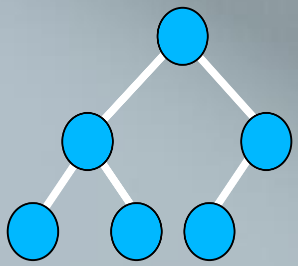
- **Árvores e suas representações**
  - Como a árvore é acíclica e conexa, existe somente **um** caminho da raiz para qualquer outro nó da árvore.
  - A **profundidade** de um nó é o comprimento do caminho da raiz ao nó.
  - A **altura** de uma árvore é a maior profundidade dos nós na árvore.
  - Um nó sem filhos é chamado de **folha** da árvore.
  - Uma **floresta** é uma coleção de árvores disjuntas.



Grafos e Árvores

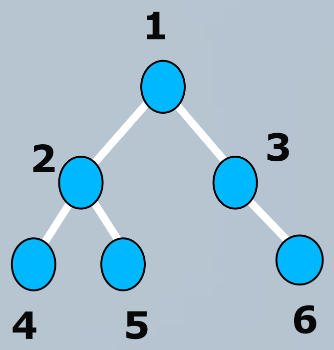
- **Árvores e suas representações**
  - As **árvores binárias** são as que cada nó tem, no máximo, dois filhos (esquerdo e direito).
  - **Árvore binária cheia** é uma árvore com todos os nós internos com dois filhos e todas as folhas estão à mesma profundidade.
  - **Árvore binária completa** é uma árvore binária quase cheia, o nível mais baixo vai se completando da esquerda para direita, mas pode ter folhas faltando.



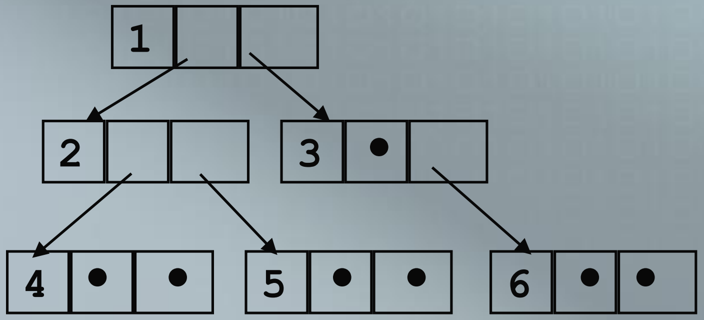


Grafos e Árvores

- **Árvores e suas representações**
  - Como um árvore também é um grafo, as representações de grafos podem ser usadas para árvores.
  - Árvores binárias têm características especiais na representação, tal como a identidade dos filhos esquerdo e direito.
  - O equivalente à matriz de adjacência é uma tabela onde os contém os dados de cada nó.
  - O equivalente de uma lista de adjacência é uma coleção de registros com três campos contendo o nó em questão, um ponteiro para registro de cada nó filho.

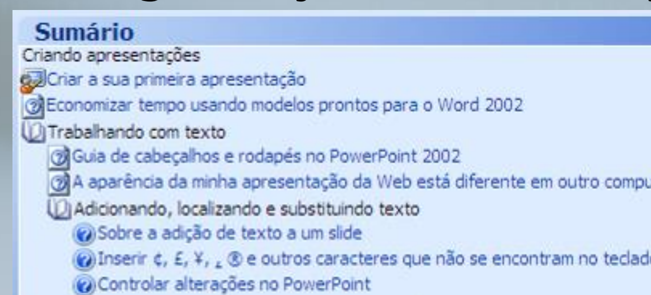


NÓ	FILHO ESQ	FILHO DIR
1	2	3
2	4	5
3	0	6
4	0	0
5	0	0
6	0	0

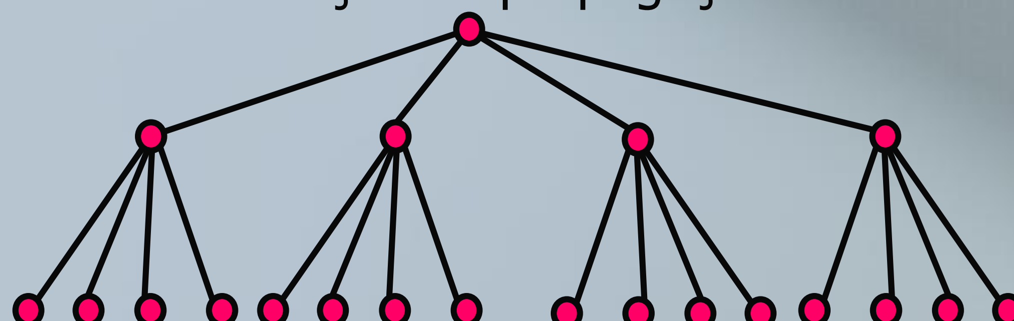


Grafos e Árvores

- **Árvores e suas aplicações**
  - Árvores genealógicas
  - Fluxo organizacional
  - Estrutura de organização de informações



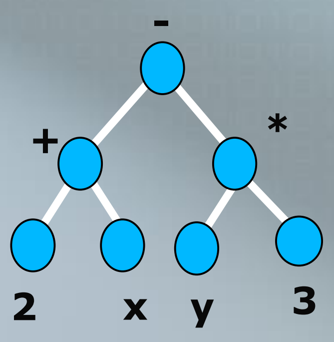
  - Demonstração de propagação de informação



$N = 4^n$

Grafos e Árvores

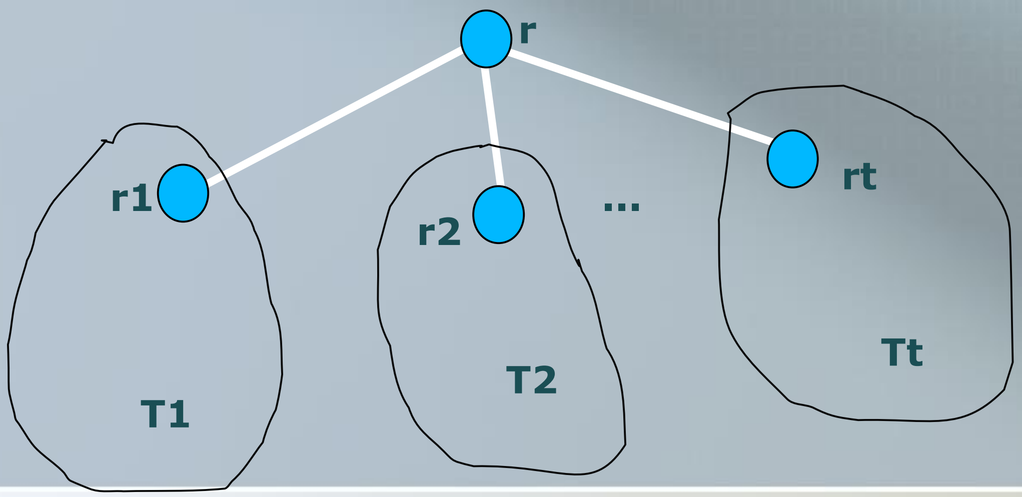
- **Árvores e suas aplicações**
  - Expressões algébricas envolvendo operações podem ser representadas por árvores algébricas rotuladas.
  - Para qualquer nó interno, a operação binária de seu rótulo é efetuada com as expressões associadas às sub-árvores.
  - Ex.: (2+x) – (y*3)



- Qual a árvore que representa a expressão (2+3) * 5?

Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - Se uma estrutura de árvore está sendo usada para armazenar dados, é útil termos um mecanismo sistemático de escrita de dados nos nós;
  - Isso pode ser feito percorrendo-se a árvore, visitando-se todos os nós na sua estrutura;
  - Os três algoritmos mais comuns de percurso em árvores são os percursos em **pré-ordem**, em **ordem simétrica** e em **pós-ordem.**
    - Seja uma árvore T com uma raiz r, com sub-árvores da esquerda para a direita, T1, T2.. Tt.



Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - Os termos **pré-ordem**, em **ordem simétrica** e em **pós-ordem**, referem-se à ordem da visita da raiz em comparação com os nós das sub-árvores.
  - No percurso em pré-ordem, **a raiz é visitada primeiro** e depois **processam-se as sub-árvores, da esquerda para a direita**, cada uma em pré-ordem.

ALGORITMO Pré-Ordem

Pré-ordem(árvore T)

```
//Escreve os nós de uma árvore com raiz r em pré-ordem
    escreva (r)
    para i=1 até t faça
        Pré-ordem (Ti)
    fim do para

fim Pré-ordem
```

Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - No percurso em **ordem simétrica**, **a sub-árvore da esquerda** é percorrida em **ordem simétrica,** depois **a raiz é visitada e,** em seguida**,** as outras **sub-árvores, da esquerda para a direita,** sempre em ordem simétrica.
    - Se a árvore for binária, a raiz é visitada entre as duas sub-árvores.

ALGORITMO OrdemSimétrica

OrdemSimétrica(árvore T)

//Escreve os nós de uma árvore com raiz r em ordem simétrica

    OrdemSimétrica(T_1)
    escreva (r)
    para *i*=2 até t **faça**
        OrdemSimétrica (T*i*)
    fim do para

fim OrdemSimétrica

Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - No percurso em **pós-ordem**, **a raiz é a última a ser visitada ,** após o percurso, em pós-ordem**,** de todas **as sub-árvores da esquerda para a direita.**

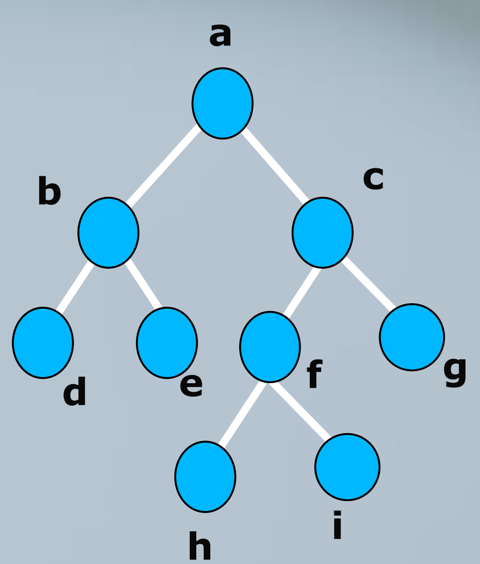
ALGORITMO Pós-Ordem

Pós-ordem(árvore T)
//Escreve os nós de uma árvore com raiz r em pós-ordem
    para *i*=1 até t **faça**
        Pós-ordem (T*i*)
    fim do para
    escreva (r)

fim Pós-ordem

Grafos e Árvores

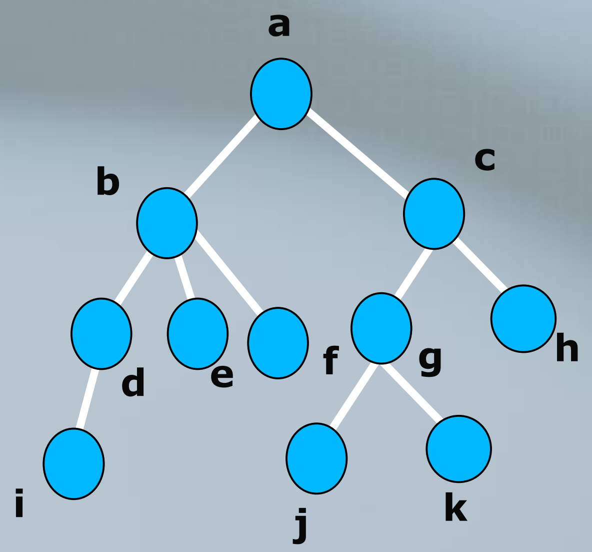
- **Algoritmos de percurso em Árvores**
  - Em árvores binárias:
    - Pré-ordem: raiz, esquerda, direita
    - Ordem simétrica: esquerda, raiz, direita
    - Pós-ordem: esquerda, direita, raiz.



- Pré-ordem: *a b d e c f h i g*
- Ordem simétrica: *d b e a h f i c g*
- Pós-ordem: *d e b h i f g c a*

Grafos e Árvores

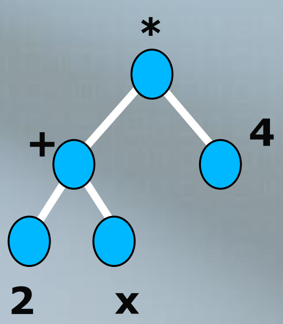
- **Algoritmos de percurso em Árvores**
  - Escreva os percursos em pré-ordem, ordem simétrica e pós-ordem da árvore abaixo:



- Pré-ordem: *a b d i e f c g j k h*
- Ordem simétrica: *i d b e f a j g k c h*
- Pós-ordem: *i d e f b j k g h c a*

Grafos e Árvores

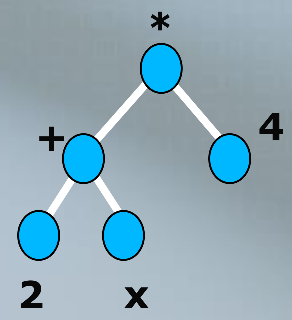
- **Algoritmos de percurso em Árvores**
  - Vimos que expressões algébricas podem ser representadas por árvores binárias.
  - Se fizermos um percurso em ordem simétrica na árvore abaixo, obteremos a expressão (2+x) * 4 – Notação infixa.



  - Um percurso em pré-ordem fornece a expressão *+ 2 x 4
    - O símbolo precede o operando.
    - Essa forma de expressão é chamada de **notação prefixa** ou **notação polonesa.**
      - * + 2 x 4 → * (2 + x) 4 → (2 + x) * 4

Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - Um percurso em pós-ordem fornece a expressão 2 x + 4*
    - O símbolo vem após os operandos.
    - Essa forma de expressão é chamada de **notação posfixa** ou **notação polonesa reversa (NPR).**
      - 2 x + 4 * → (2 + x) 4 * → (2 + x) * 4



  - Embora pouco familiares, essas notações dispensam parênteses para evitar ambiguidades e são mais eficientes.
  - Compiladores normalmente mudam expressões algébricas de programas para NPR para obter processamento mais eficiente.

Grafos e Árvores

- **Algoritmos de percurso em Árvores**
  - Exercício: Escreva a árvore que representa a expressão:

    a + (b * c – d) e escreva a expressão em notações polonesa e polonesa reversa.