



Prof. Thiago Jabur Bittar

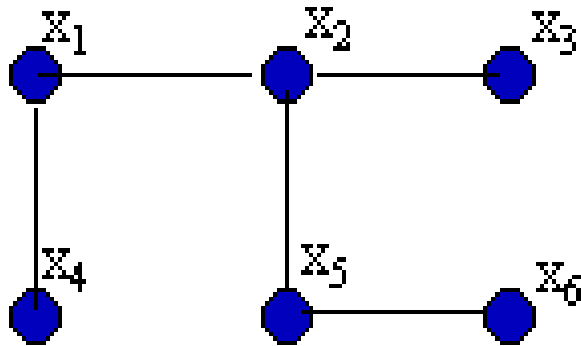
Teoria dos Grafos

Árvores

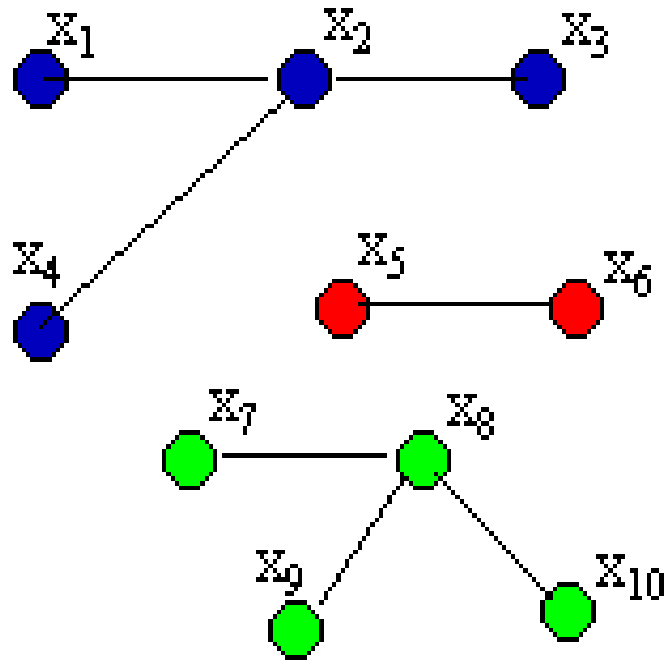
Árvores

Uma **árvore** é um grafo conexo não orientado e sem circuitos

Se existe um **caminho** entre cada par de vértices distintos do grafo.



Uma **floresta** é um grafo cujas componentes conexas são árvores

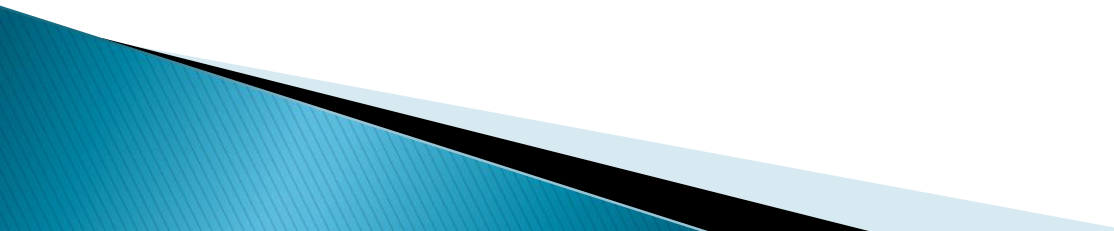


Teorema

Um grafo não orientado é uma árvore se e somente se existe um único caminho simples entre qualquer par de vértices.

Prova

Assuma que G é uma árvore. Logo G é um grafo conexo e sem circuitos simples. Sejam x e y dois nós de G . Logo, como G é conexo, existe um caminho simples entre x e y . Adicionalmente, esse caminho é único, pois se existisse um outro caminho, o caminho formado através da combinação do caminho de x até y com o segundo caminho começando por y e chegando a x formaria um circuito, o que contraria a hipótese de que G é uma árvore.

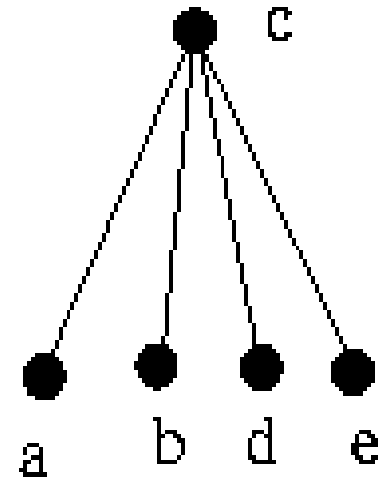
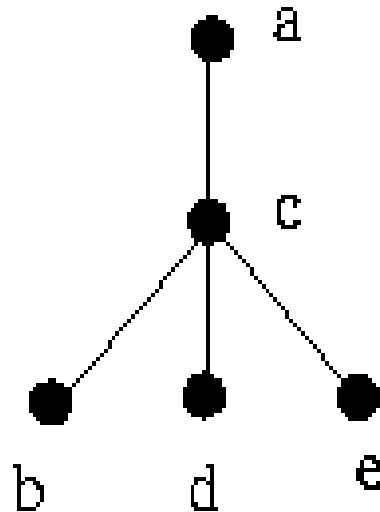
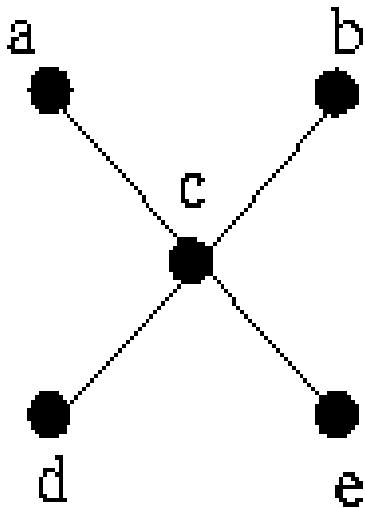


Teorema

Uma árvore com n nós possui $n-1$ arestas.

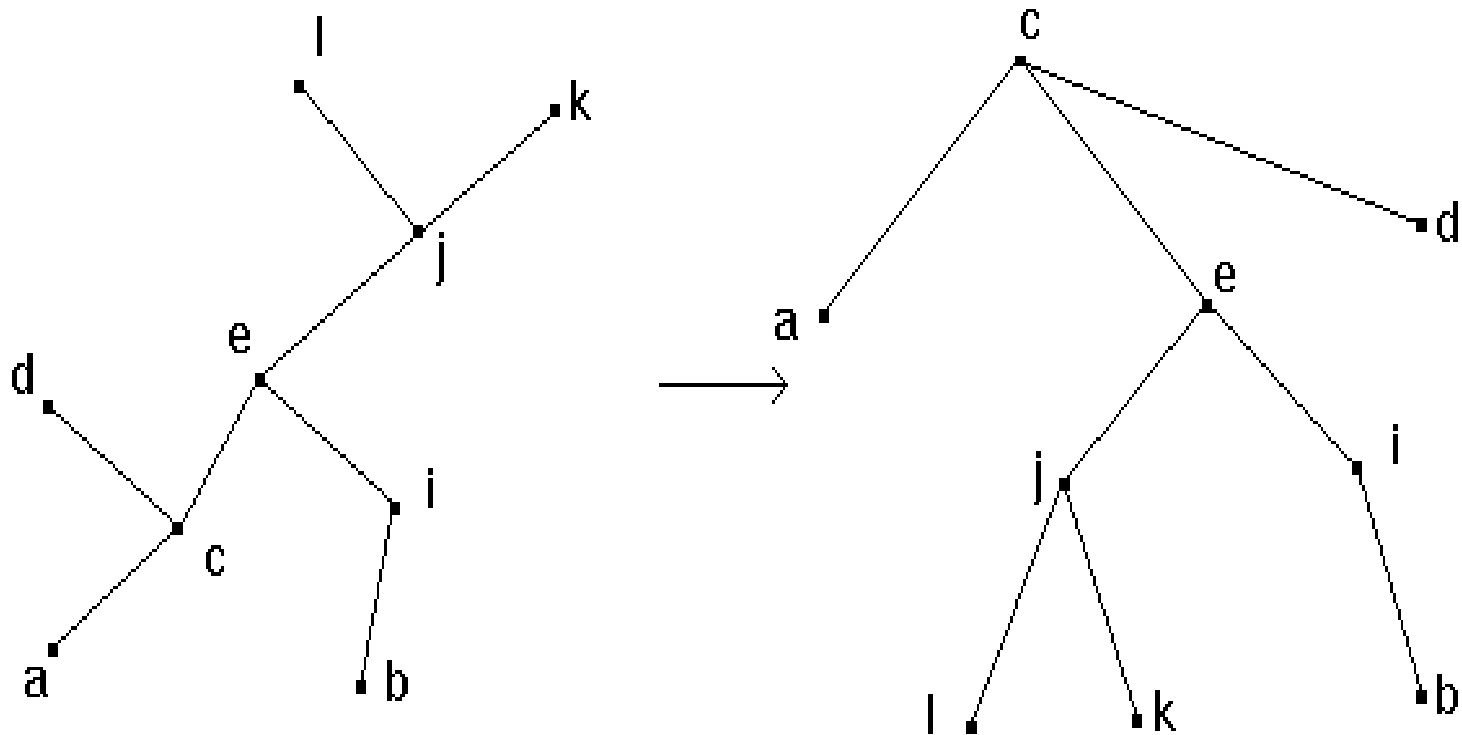
Árvore Enraizada

Uma árvore $T = (V, E)$ é denominado **enraizada** quando algum vértice v é escolhido como especial. Esse vértice v é a **raiz** da árvore.



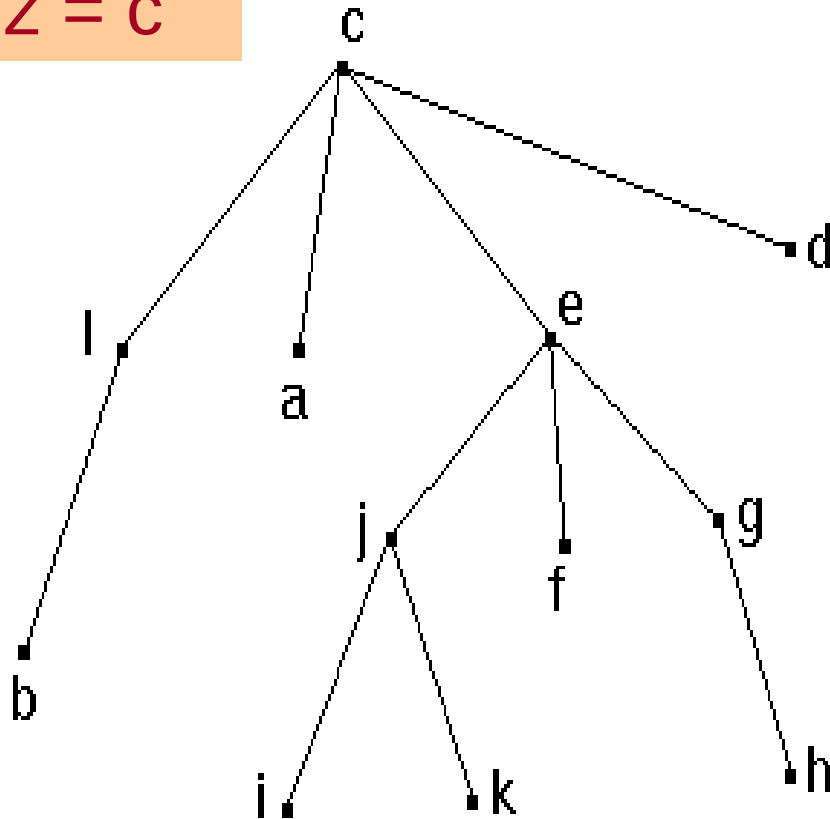
Árvore Enraizada

Usualmente representamos graficamente a raiz no topo. Podemos transformar uma árvore sem raiz numa árvore enraizada simplesmente escolhendo um vértice como raiz.



Árvore Enraizada

Raiz = c



ancestrais de $j=\{e,c\}$

descendentes de $j=\{i,k\}$

pai de $j=e$

filhos de $j=\{i,k\}$

nível de $j=2$

altura da árvore = 3

folhas = $\{b,a,i,k,f,h,d\}$

Árvore Enraizada

A raiz de uma árvore não possui pai, e todo vértice v diferente de r , possui um único pai.

Uma **folha** é um vértice que não possui filhos. Vértices que possuem filhos são chamados de **vértices internos**.

Quando a raiz é o único nó do grafo ela é uma **folha**.

O **nível** da raiz é zero, de seus filhos é 1. O nível de um nó é igual ao nível de seu pai mais um.

Para dois vértices irmãos v e w , $\text{nível}(v) = \text{nível}(w)$.

A **altura** de uma árvore é o valor máximo de **nível**(r) para todo vértice v de T .

Subárvore

Seja $T(V,E)$ uma árvore enraizada e v pertencente a V

Uma subárvore T_v de T é uma árvore enraizada cuja raiz é v , definida pelo subgrafo induzido pelos descendentes de v mais o próprio v .

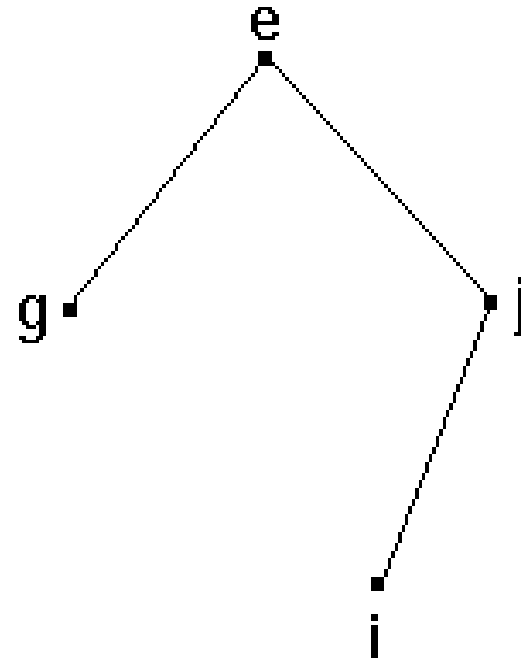
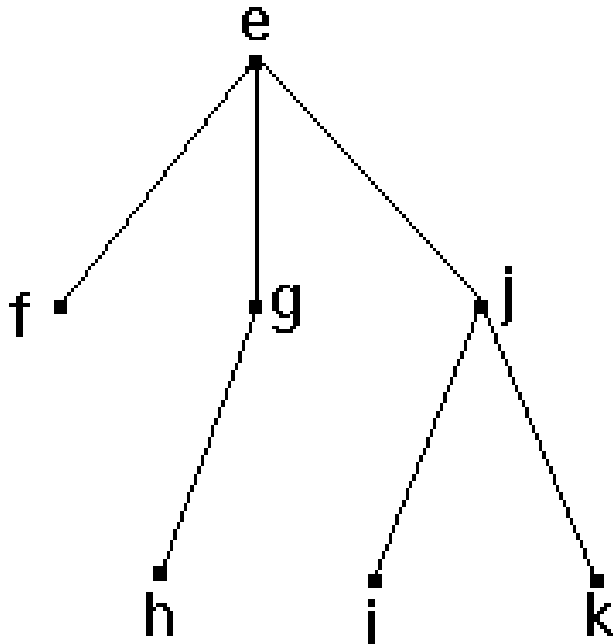
A subárvore de raiz v é única para cada v pertencente a V .

Árvore m-ária

Uma árvore enraizada é chamada de **m-ária** se todo nó interno não possui mais que **m** filhos.

A árvore é chamada **árvore m-ária completa** se todo nó interno possuir exatamente **m** filhos.

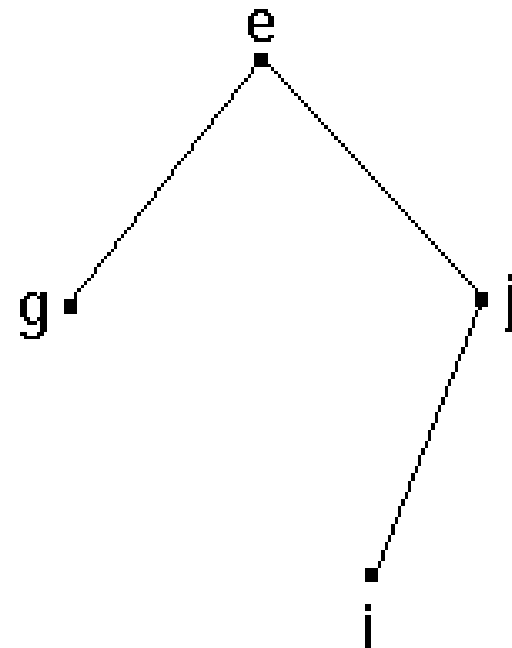
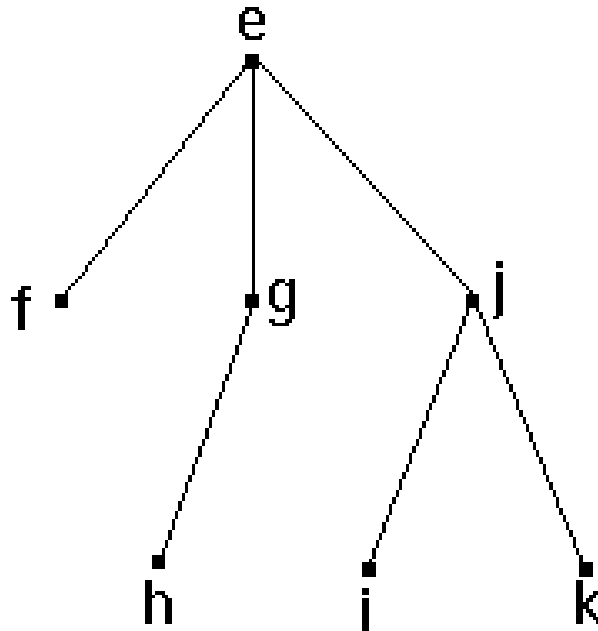
Uma árvore m-ária com $m=2$ é chamada de **árvore binária**.



Árvore m-ária

Uma árvore enraizada m-ária de altura h é **balanceada** se todas as folhas estão no nível h ou $h-1$.

É **balanceada completa** se todas as folhas estão no nível h .



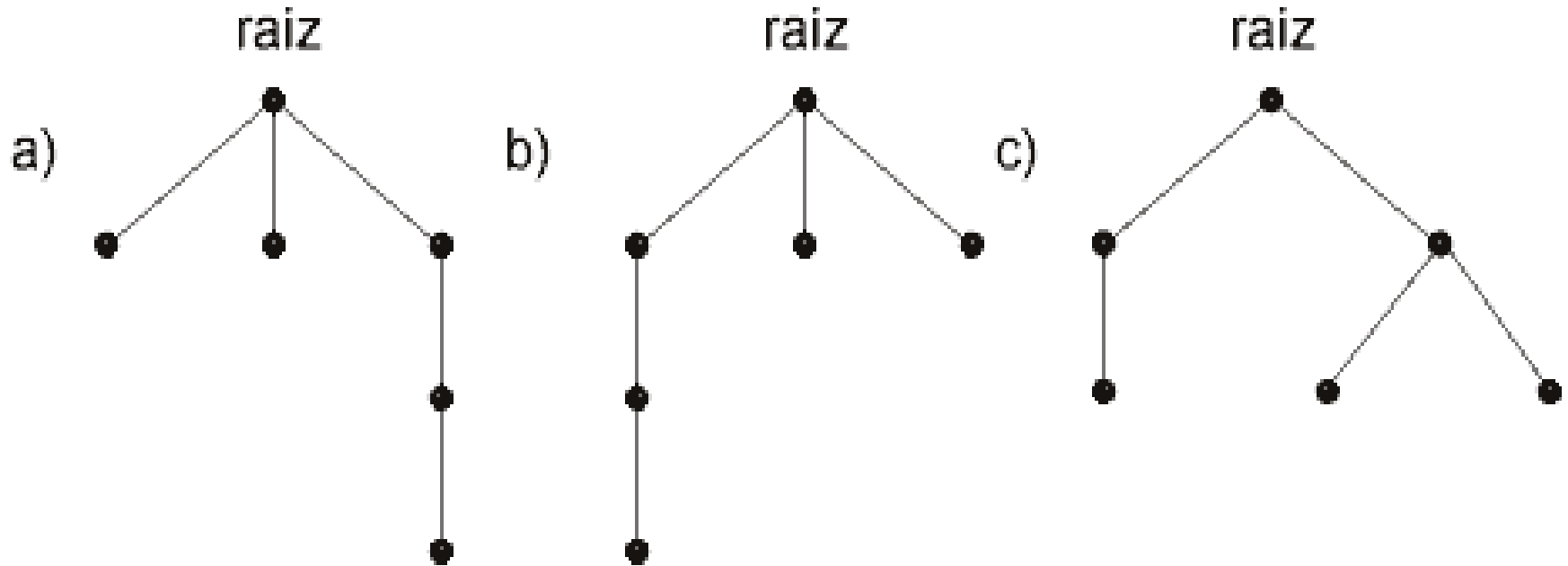
Árvore Enraizada Ordenada

Na definição de árvore enraizada, é irrelevante a ordem em que os filhos de cada vértice v são considerados.

Caso a ordenação seja relevante a árvore é denominada **enraizada ordenada**.

Assim, para cada vértice v pode-se identificar o primeiro filho de v (o mais a esquerda), o segundo filho (o segundo mais a esquerda), etc.

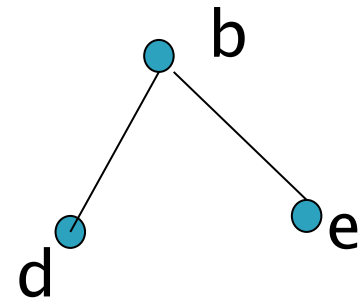
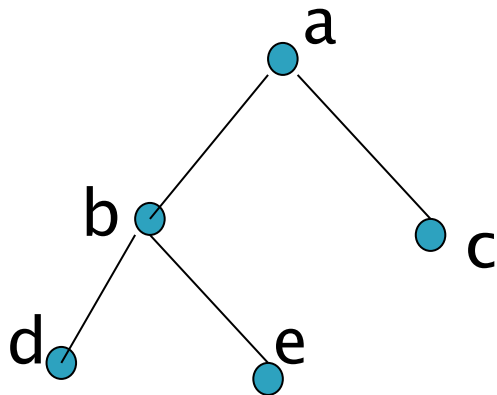
Árvores



Árvore enraizada ordenada

No caso de árvores binárias, se um nó interno possui dois filhos, temos o **filho da esquerda** e o **filho da direita**

A árvore cuja raiz é o filho da esquerda de um vértice é chamada de **subárvore da esquerda** desse vértice.

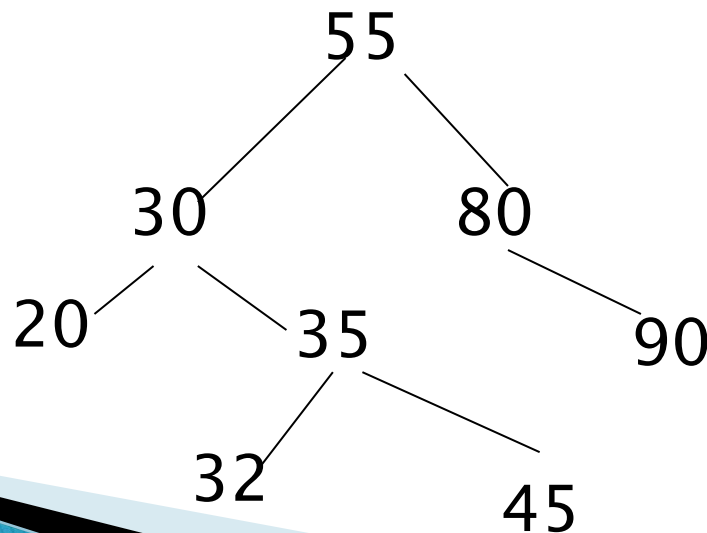


Subárvore da esquerda
de a

Árvore binária de busca

Busca de itens numa lista.

Cada vértice é rotulado por uma **chave** de forma que a chave de um vértice é maior do que as chaves de todos os nós da subárvore da esquerda e menor do que as chaves dos nós da subárvore da direita.



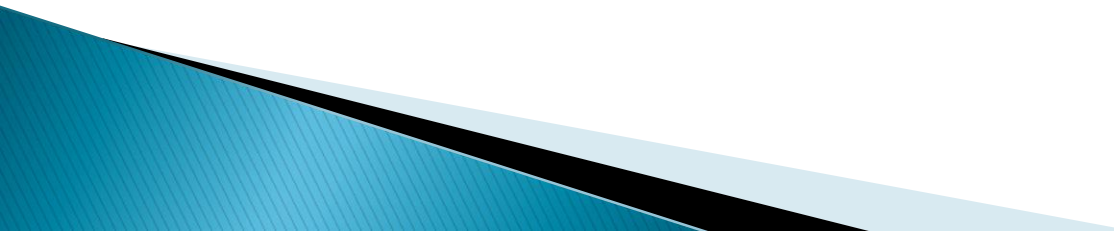
Construindo uma árvore binária de busca

Procedimento recursivo que recebe uma lista de itens.

O primeiro item da lista é a raiz da árvore.

Para adicionar um novo item compare-o com os nós que já estão na árvore: comece pela raiz e siga para a esquerda se o item é menor que o item que rotula o nó que está sendo comparado ou siga para a direita, caso contrário.

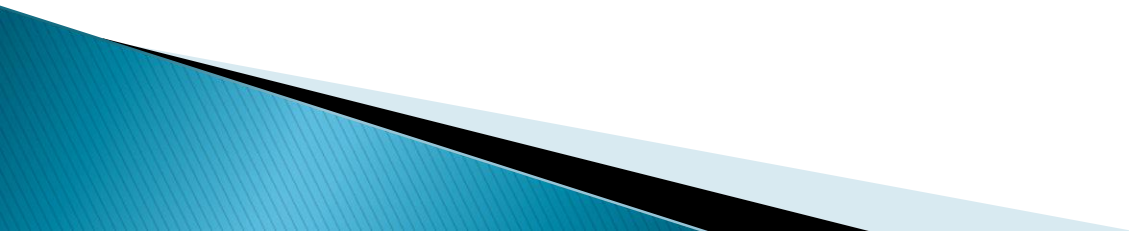
Quando o novo item é menor que um item cujo nó não tem filho da esquerda, adicione-o como filho da esquerda desse nó. Analogamente, quando o item é maior que o item cujo nó não tem filho da direita, adicione-o como filho da direita desse nó



Construindo uma árvore binária de busca

Construa uma árvore binária de busca a partir da seguinte lista:

77,14,83,20,45,32,50,49



Caminhamento em pré-ordem

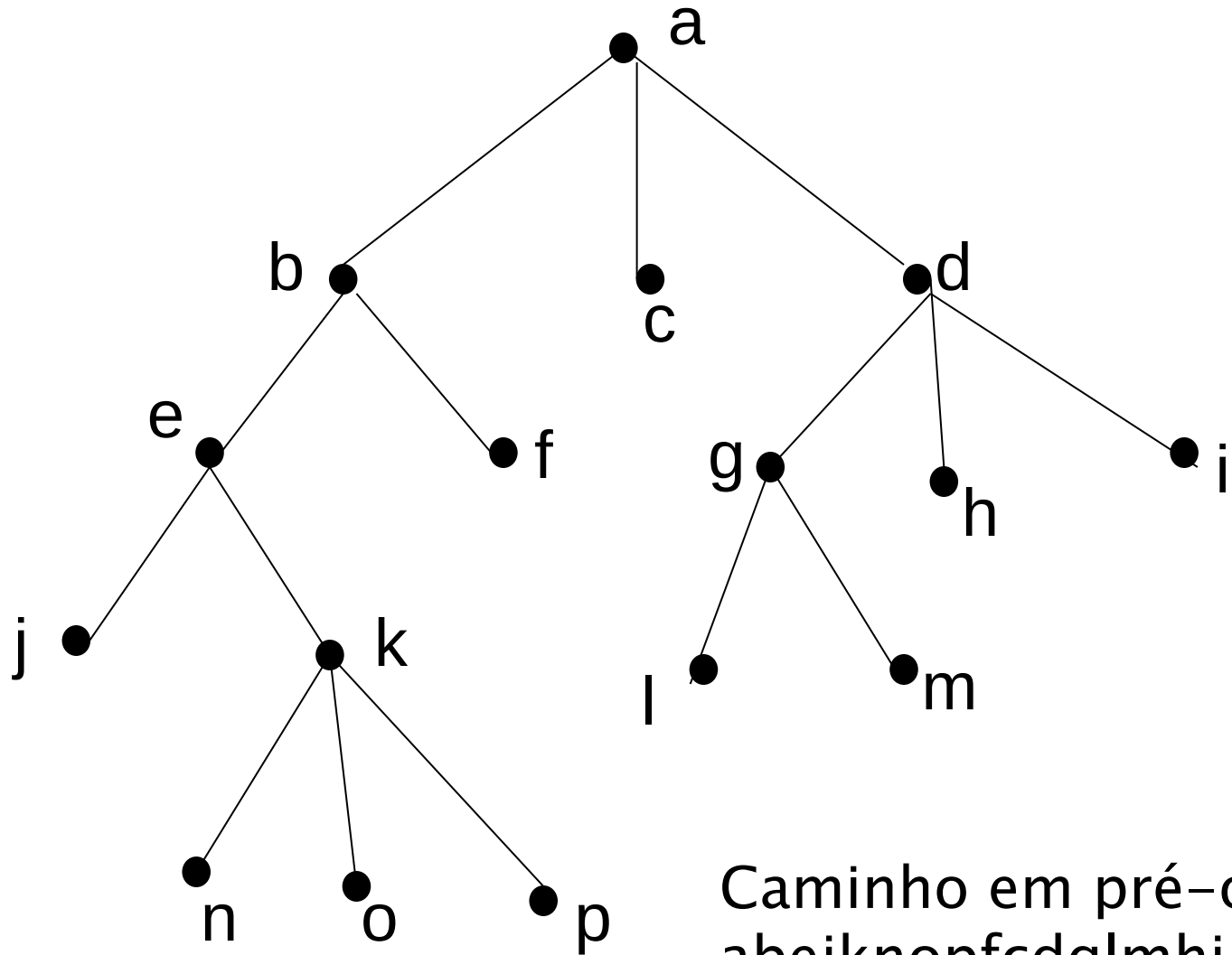
Seja T uma árvore enraizada e ordenada com raiz r .

Se T possui apenas r , então o caminho em pré-ordem de T é r .

Caso contrário, sejam $T_1, T_2, \dots, T_n$ as subárvores de r da esquerda para a direita.

O caminhamento em pré-ordem começa visitando r e continua fazendo um caminhamento em pré-ordem em T_1 , em seguida em T_2 , e assim sucessivamente até que T_n seja percorrida em pré-ordem.

Exemplo



Caminho em pré-ordem
abejknopfc dglmhi

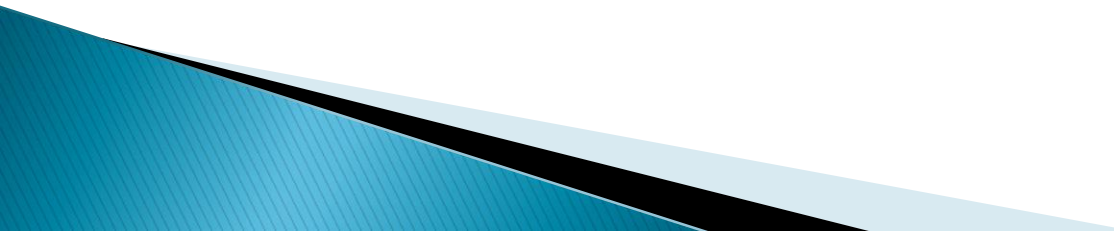
Caminhando em ordem

Seja T uma árvore enraizada e ordenada com raiz r .

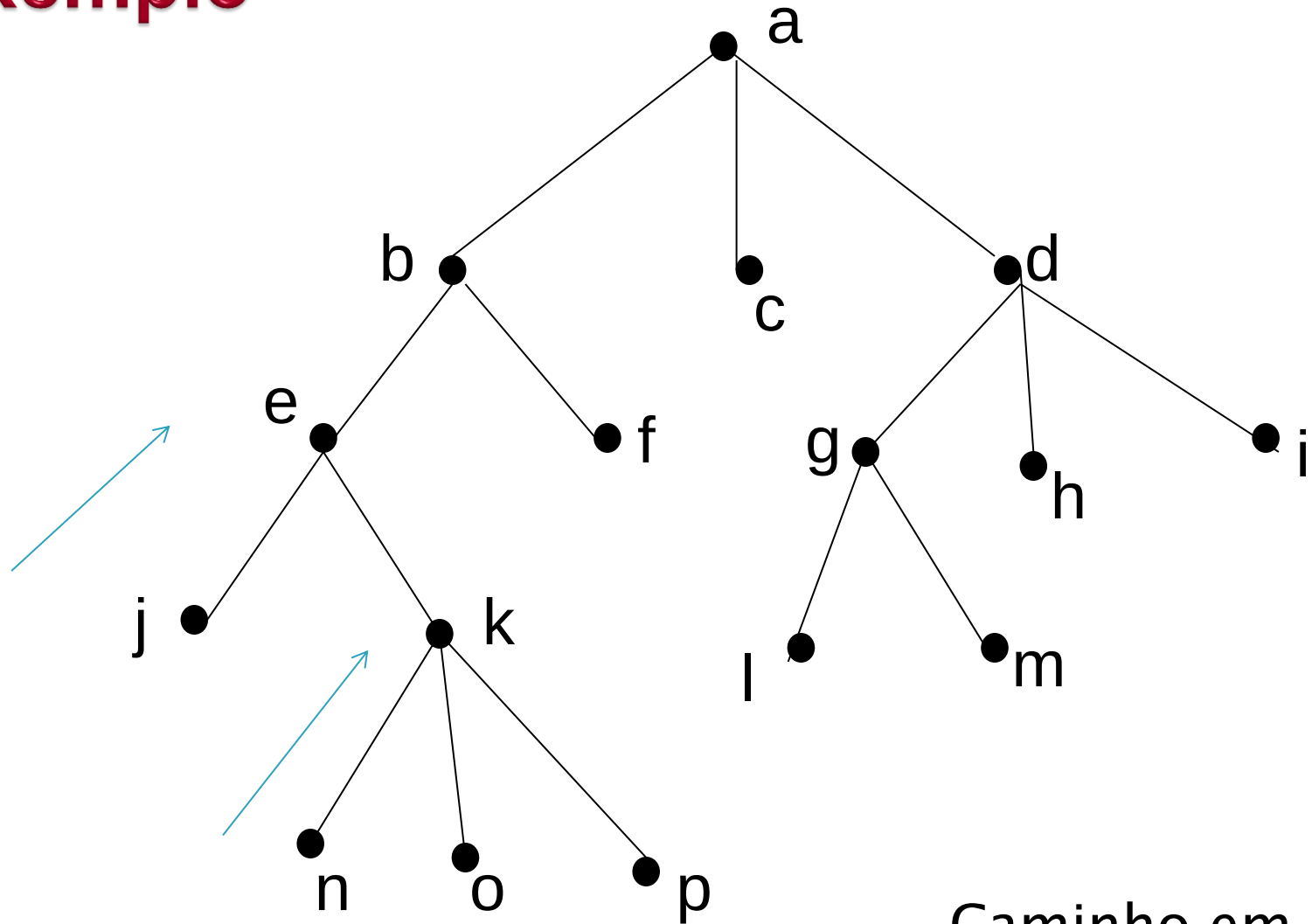
Se T possui apenas r , então o caminhamento em ordem de T é r .

Caso contrário, sejam $T_1, T_2, \dots, T_n$ as subárvores de r da esquerda para a direita.

O caminhamento em ordem começa percorrendo em ordem T_1 , em seguida visita r , e continua fazendo um caminhamento em ordem em T_2 , em T_3 , e finalmente em T_n .



Exemplo



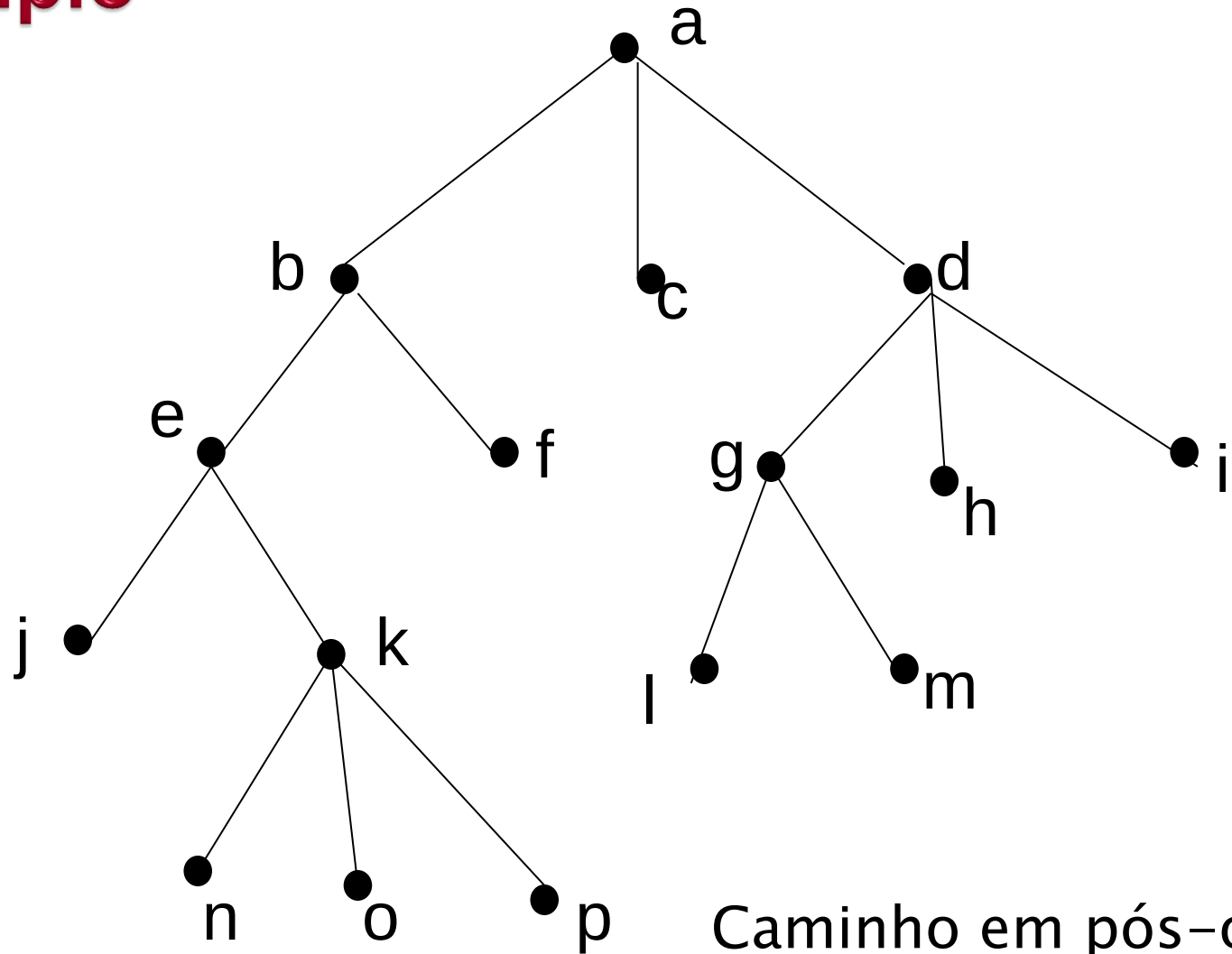
Caminho em ordem
jenkopbfaclgmdhi

Caminhando em pós-ordem

Seja T uma árvore enraizada e ordenada com raiz r .
Se T possui apenas r , então o caminho em pós-ordem de T é r .

Caso contrário, sejam $T_1, T_2, \dots, T_n$ as subárvores de r da esquerda para a direita. O caminhamento em pós-ordem começa percorrendo T_1 em pós-ordem, em seguida $T_2, T_3, \dots, T_n$, e finaliza visitando r .

Exemplo



Caminho em pós-ordem
jnopkefbclmghida