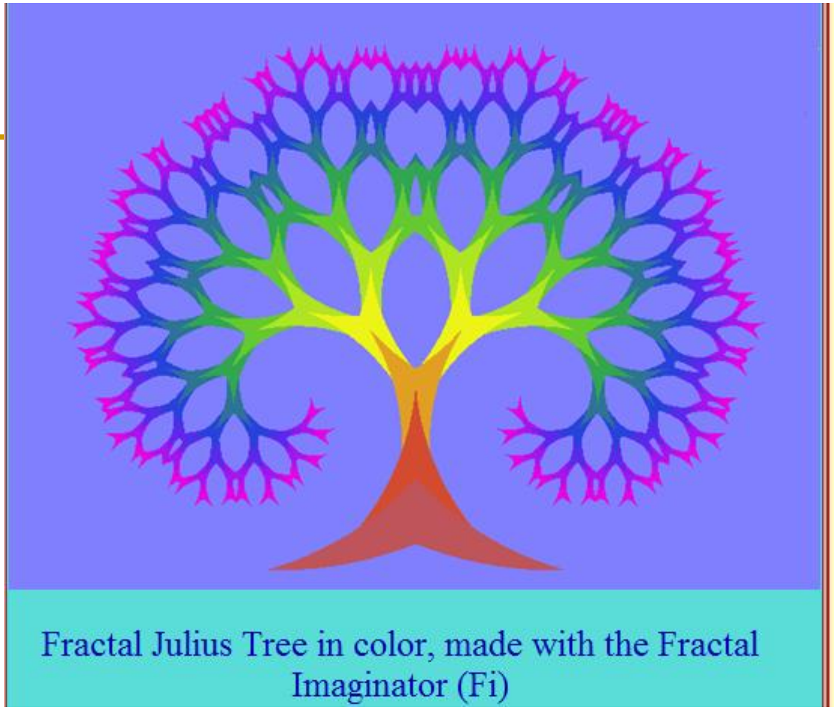
http://www.mysticfractal.com/FractalImaginator.html

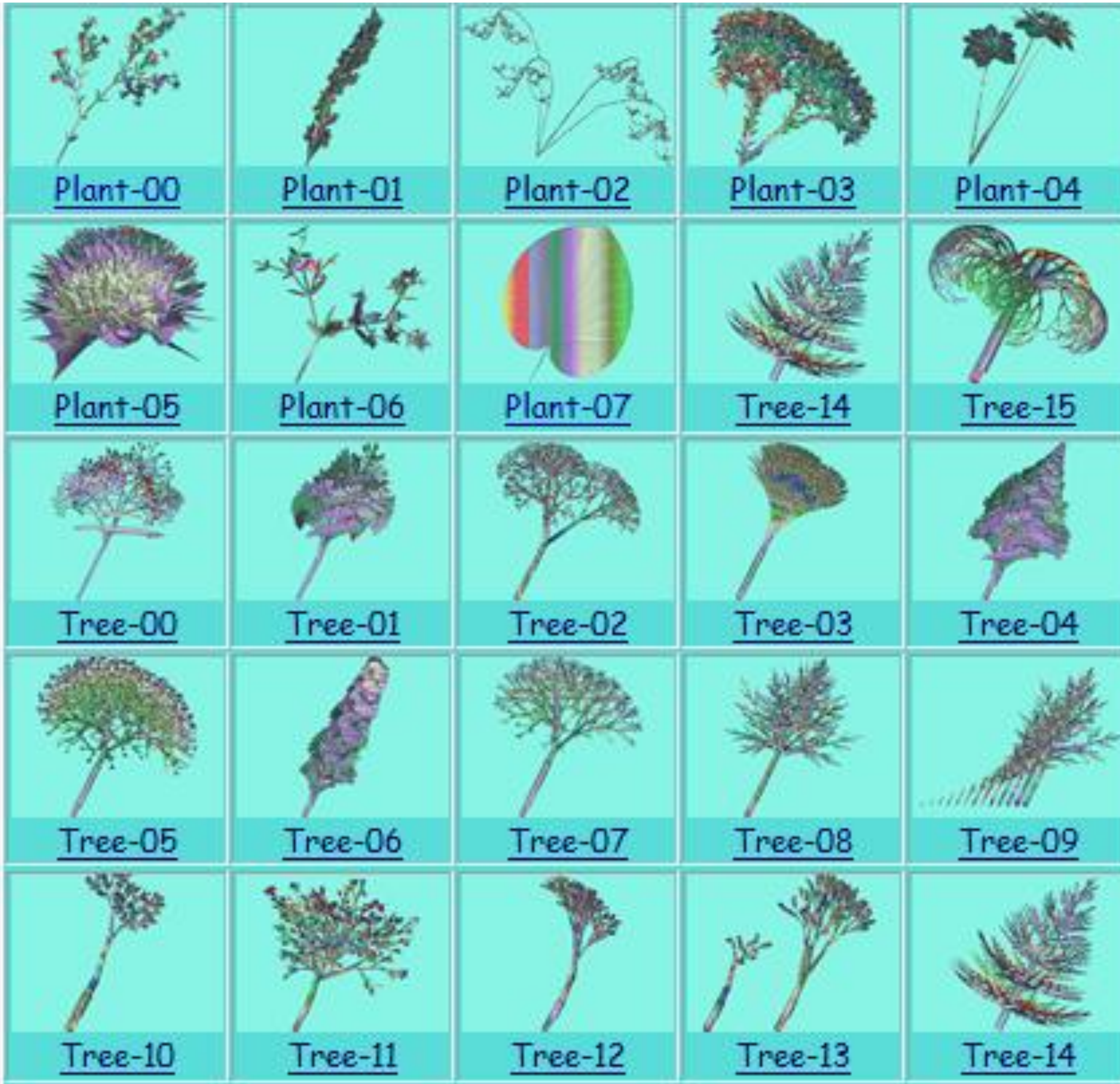


Árvores
Conceitos gerais

9/11

Nesta aula veremos conceitos e definições sobre árvores

Diferentemente das estruturas de pilhas, filas e listas que são lineares, uma **árvore** é uma estrutura de dados não linear http://www.fractal.org/Julius-Ruis-Gallery/Index-FTG.htm



Problema

- Representações/Implementações do TAD Lista Linear:
  - Lista encadeada dinâmica
    - eficiente para inserção e remoção dinâmica de elementos (início ou fim), mas ineficiente para busca (O(n))
  - Lista seqüencial (ordenada) estática
    - Eficiente para busca (busca binária), mas ineficiente para inserção e remoção de elementos (requer abrir espaços)

- Haveria uma ED que tivesse o melhor desempenho nas 3 operações?

Solução

- Árvores: solução eficiente para inserção, remoção e busca
  - Representação não linear...

- Mas as duas (listas e árvores) tem características em comum: são definidas recursivamente!

Recursão como ferramenta de definição

Uma lista do tipo T é

- Uma lista vazia ou
- Uma concatenação de um elemento do tipo **T** com uma lista cujo tipo básico também seja **T**

Uma árvore, com tipo T, é

- Uma árvore vazia ou
- Um nó do tipo **T** associado a um número finito de estruturas disjuntas de árvore do mesmo tipo **T**, denominadas **subárvores**

- Uma lista pode ser considerada como uma árvore na qual cada nó tem, no máximo, uma única subárvore.
  - Por este motivo, uma **lista** é também denominada **árvore degenerada**

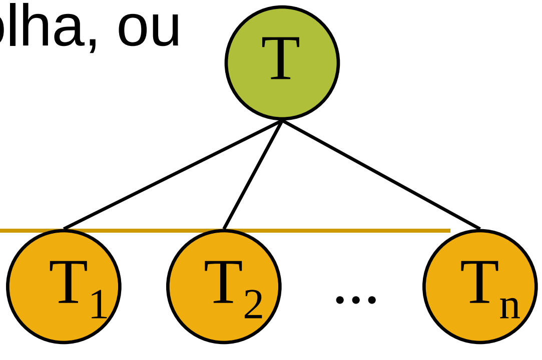
Definição Formal

- Árvore T: conjunto finito de elementos, denominados **nós** ou vértices, tais que:
  - Se T = ∅, a árvore é dita vazia; c.c.

(i) T contém um nó especial, denominado raiz;

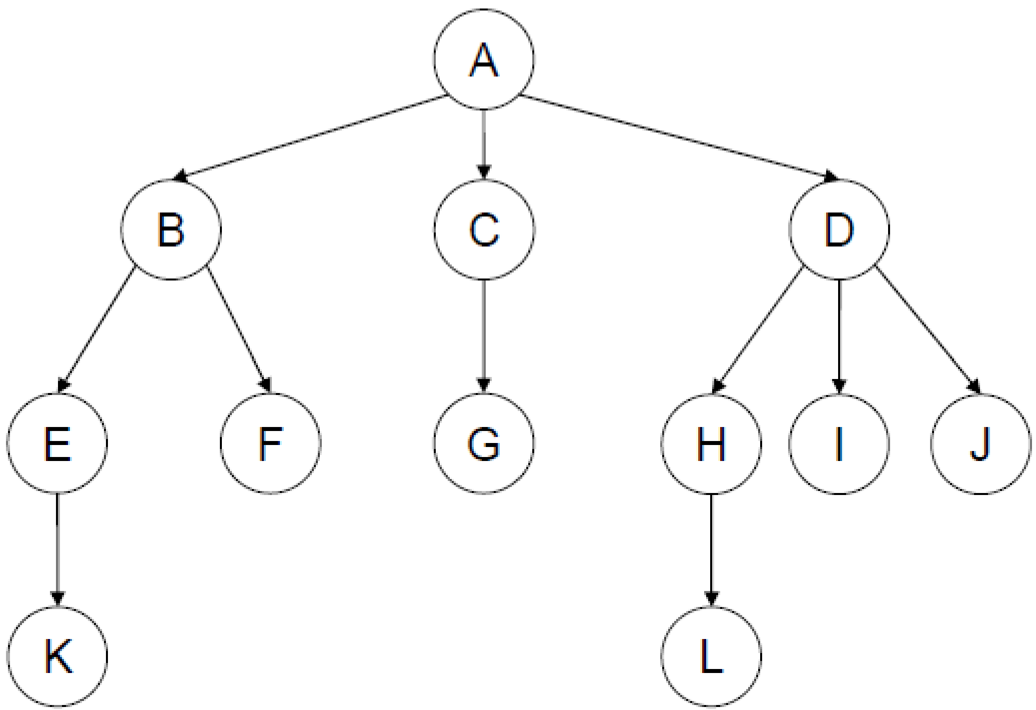
(ii) os demais nós, ou constituem um único conjunto vazio, ou são divididos em $m \geq 1$ conjuntos disjuntos não vazios ($T_1, T_2, \ldots, T_n$), que são, por sua vez, cada qual uma árvore;

- $T_1, T_2, \ldots, T_n$ são chamadas subárvores de T;
- Um nó sem subárvores é denominado nó-folha, ou simplesmente, folha



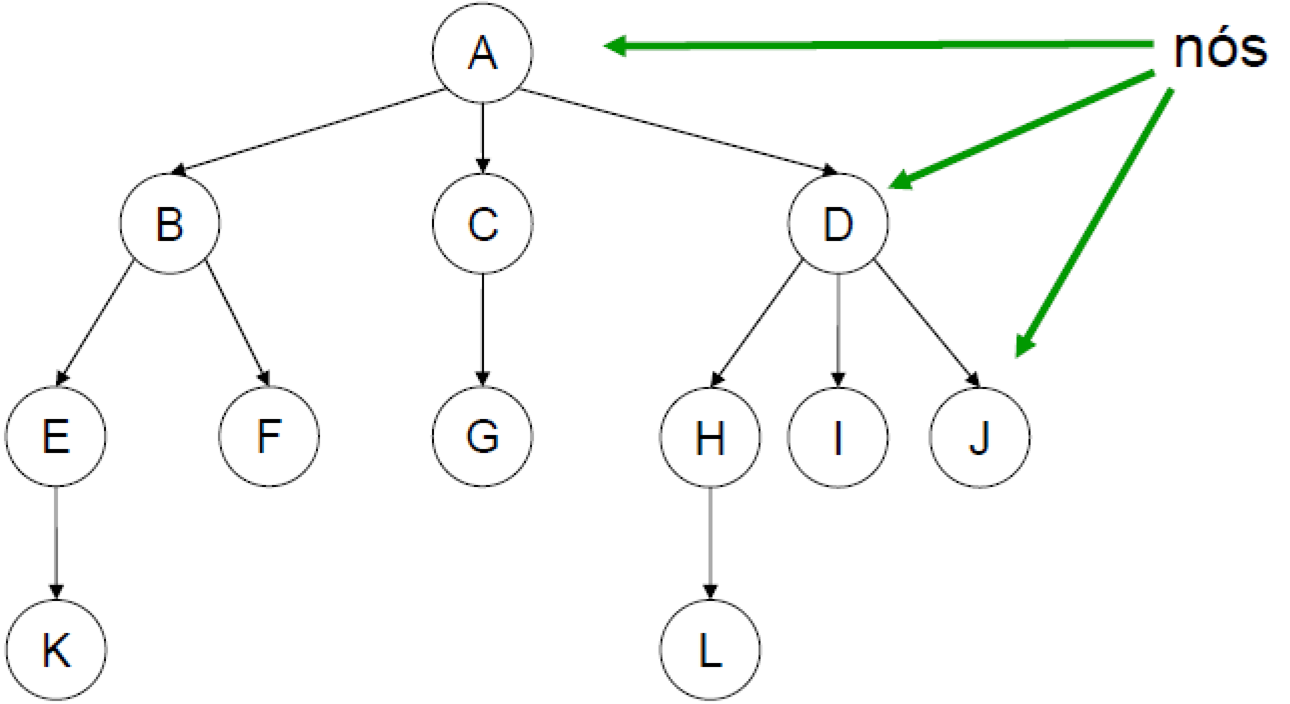
Representação

- Utilizaremos grafos para representar árvores,
  - Que são um conjunto de vértices (nós) e arestas
- Uma árvore é um grafo sem ciclos



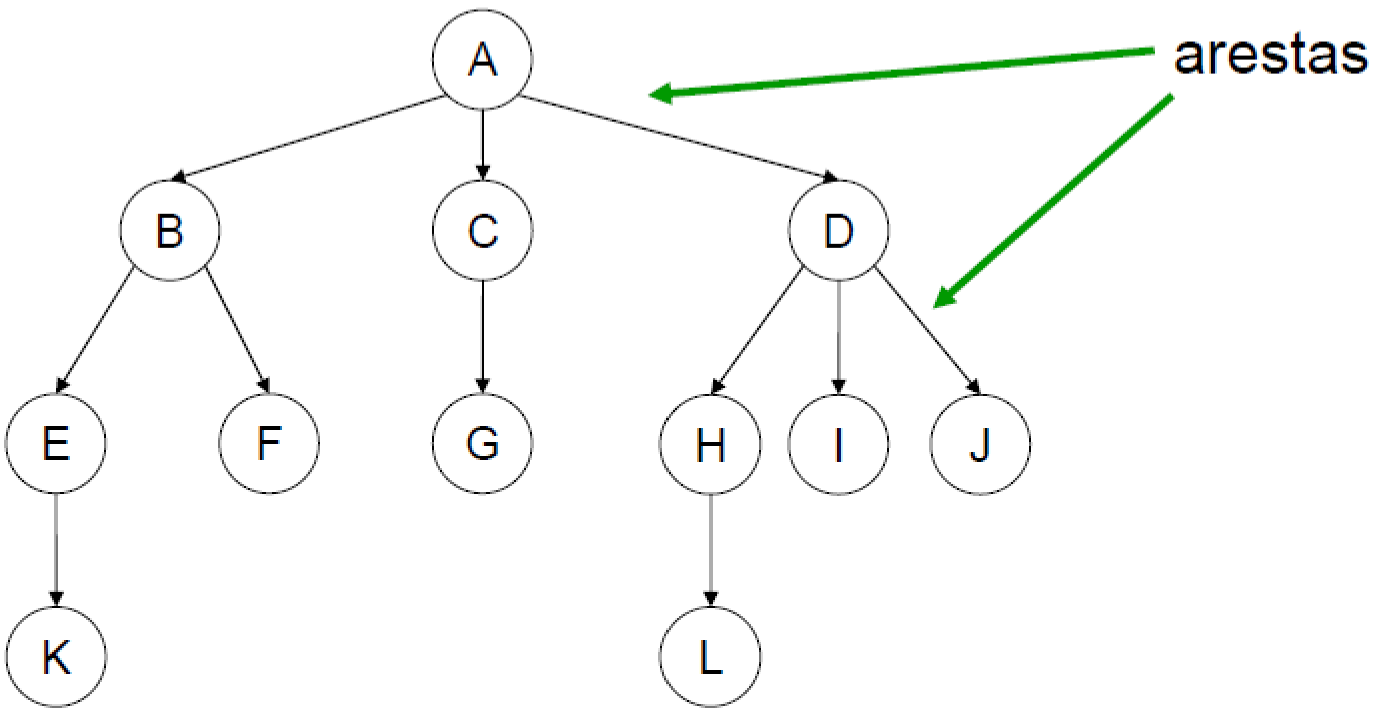
Nós (vértices)

- Esta árvore possui 12 nós (ou vértices).



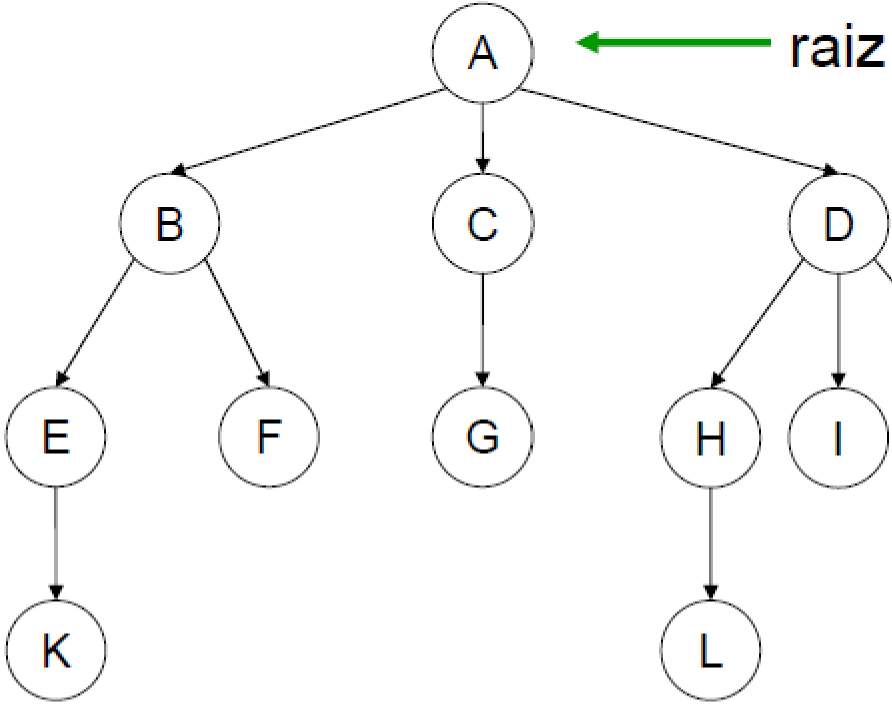
Arestas

- Uma aresta liga um nó a outro.



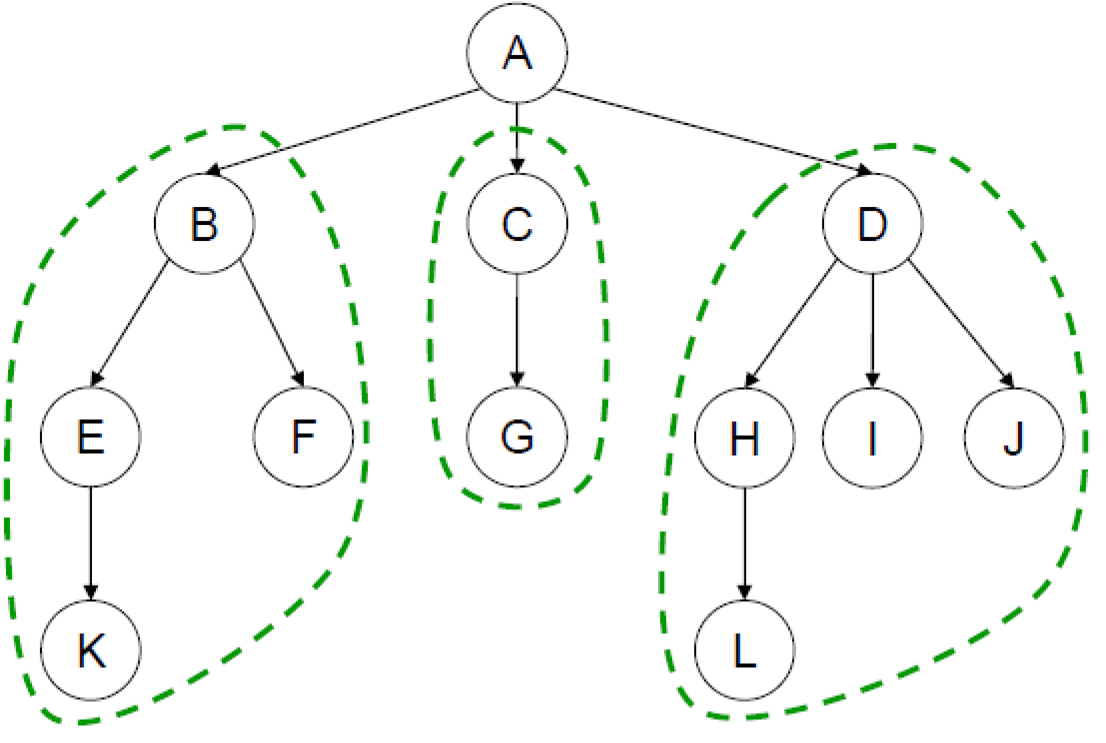
Raiz

- Normalmente, as árvores são desenhadas de forma invertida, com a raiz para cima.



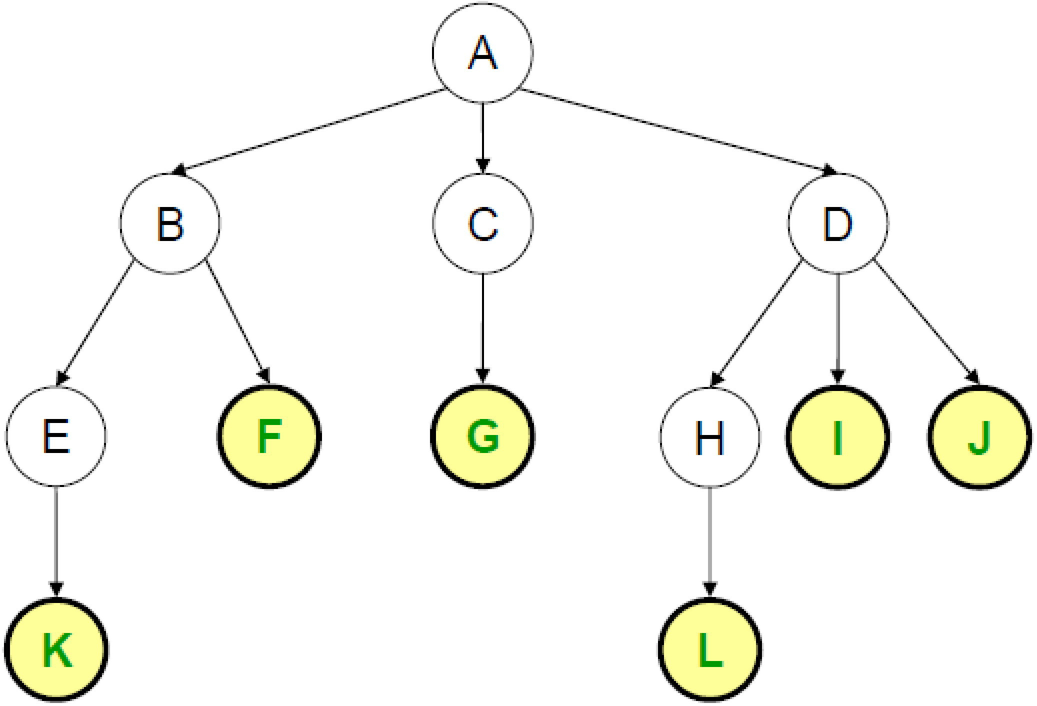
Subárvores

- O nó A possui 3 subárvores cujas raízes são B, C e D.



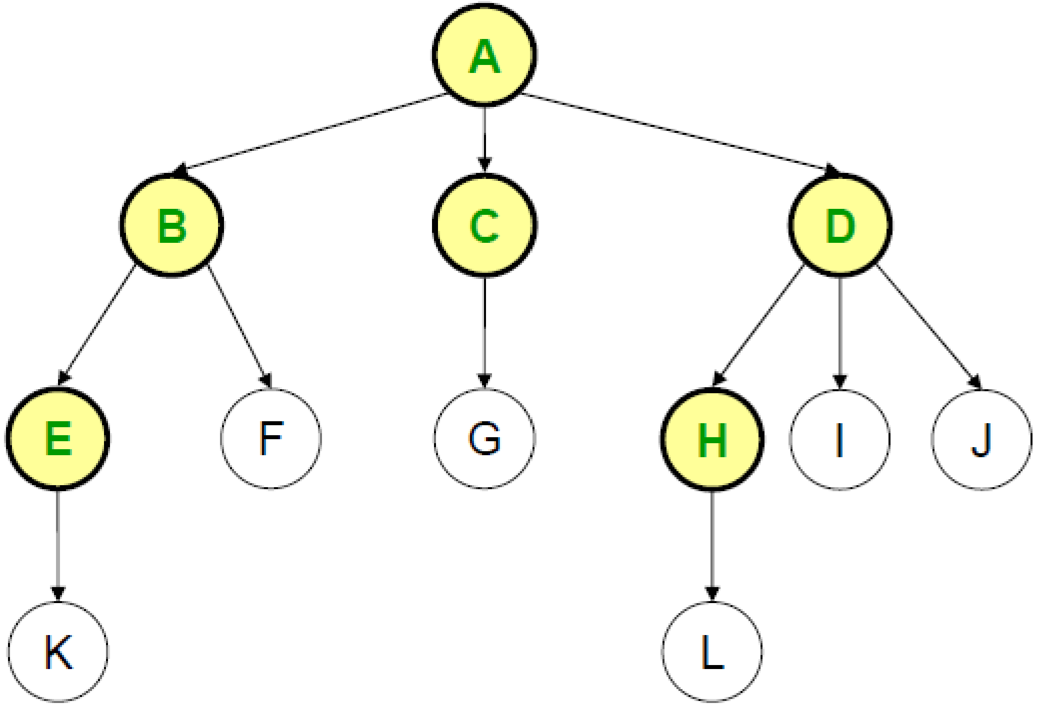
Folha

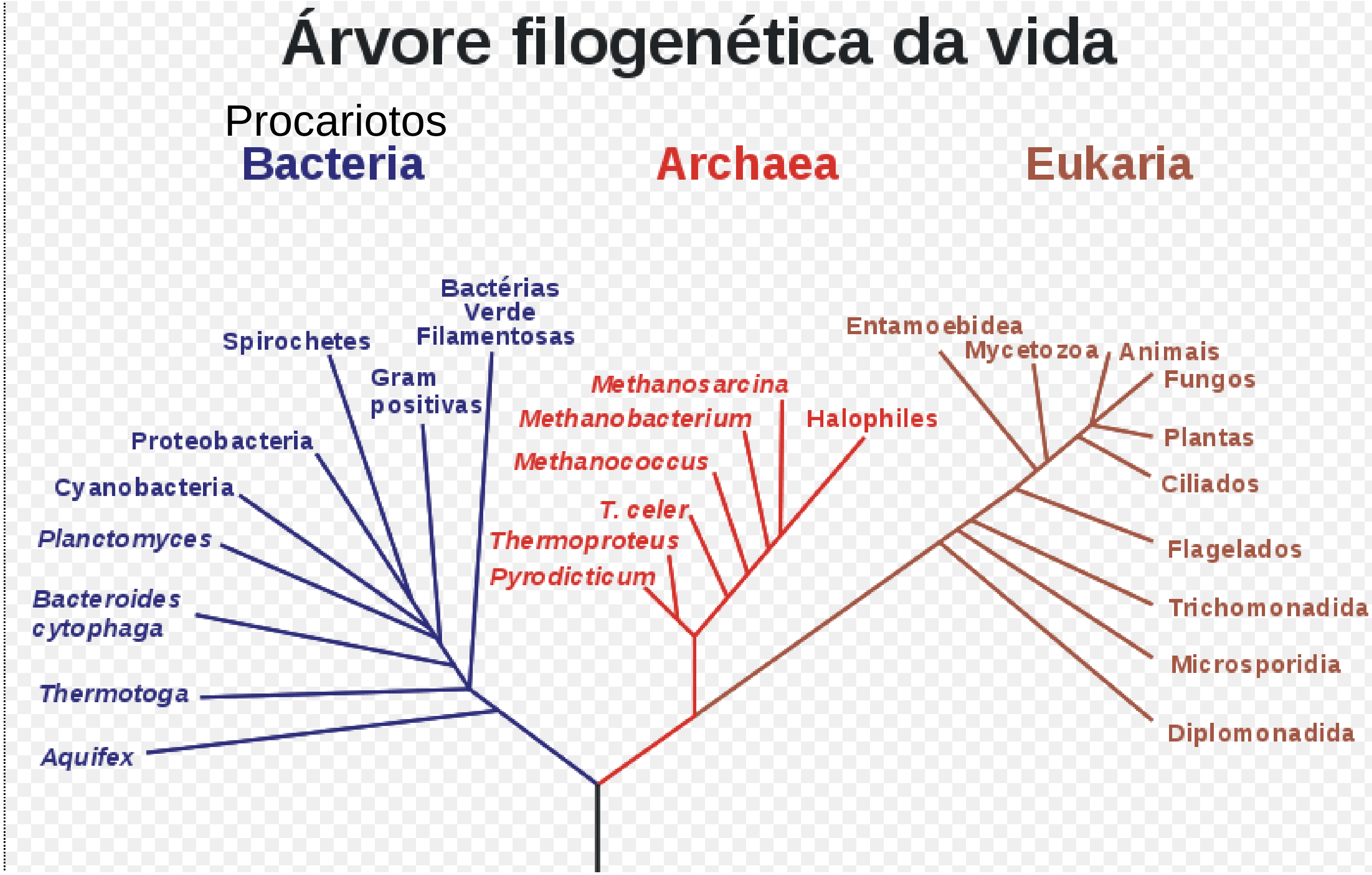
- Um nó sem descendentes (sem filhos) é denominado terminal ou folha.

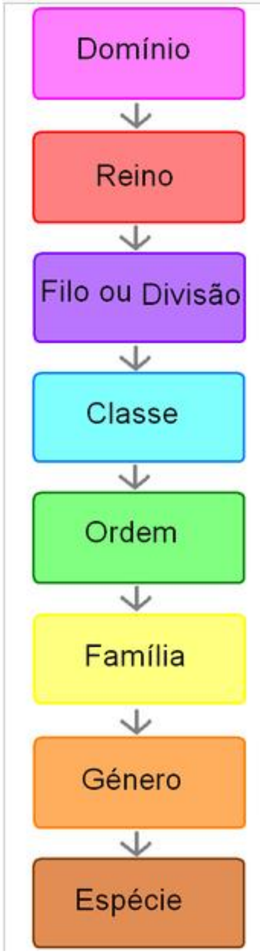


Não-Folha

- Um nó com descendentes (com filhos) é denominado não-folha ou nó interior.





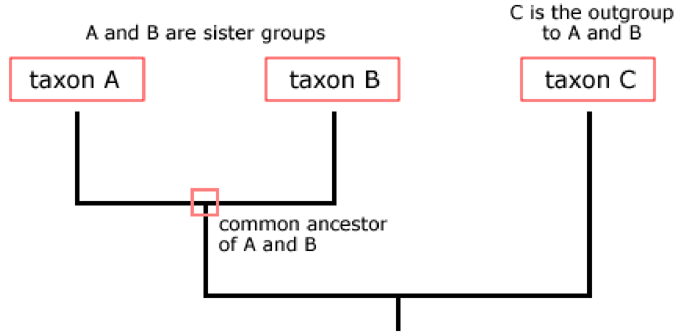


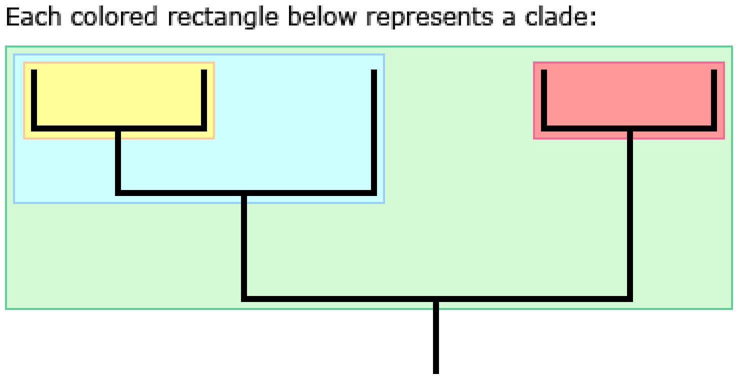
VÍRUS ???

- Árvore: adequada para representar estruturas hierárquicas não lineares, como
  - Taxonomias (espécies vivas) ou classificação biológica.
    - http://tolweb.org/tree/ (Tree of LifeWeb Project)
  - Relações de descendência (pai, filho, irmãos, etc.)

- Uma **árvore** **filogenética**, por vezes também designada por **Árvore da Vida**, é uma exibição em forma de uma árvore das relações evolutivas entre várias espécies ou outras entidades que podem ter um antepassado em comum.
- Em uma árvore filogenética, cada nodo com descendentes representa o mais recente antepassado comum, e os comprimentos dos ramos podem representar estimativas do tempo evolutivo.
- Cada nodo terminal em uma árvore filogenética é chamado uma unidade taxonômica.
- Nodos internos geralmente são chamados de Unidades Taxonômicas Hipotéticas.
- As árvores filogenéticas são confeccionadas a partir de uma matriz contendo os dados disponíveis (morfológicos, químicos ou genéticos) sobre os táxons estudados.
- Estes dados são comparados, e os táxons agrupados pelas semelhanças e diferenças entre si em clados.
- Atualmente, há alguns softwares disponíveis para a realização destes cálculos.

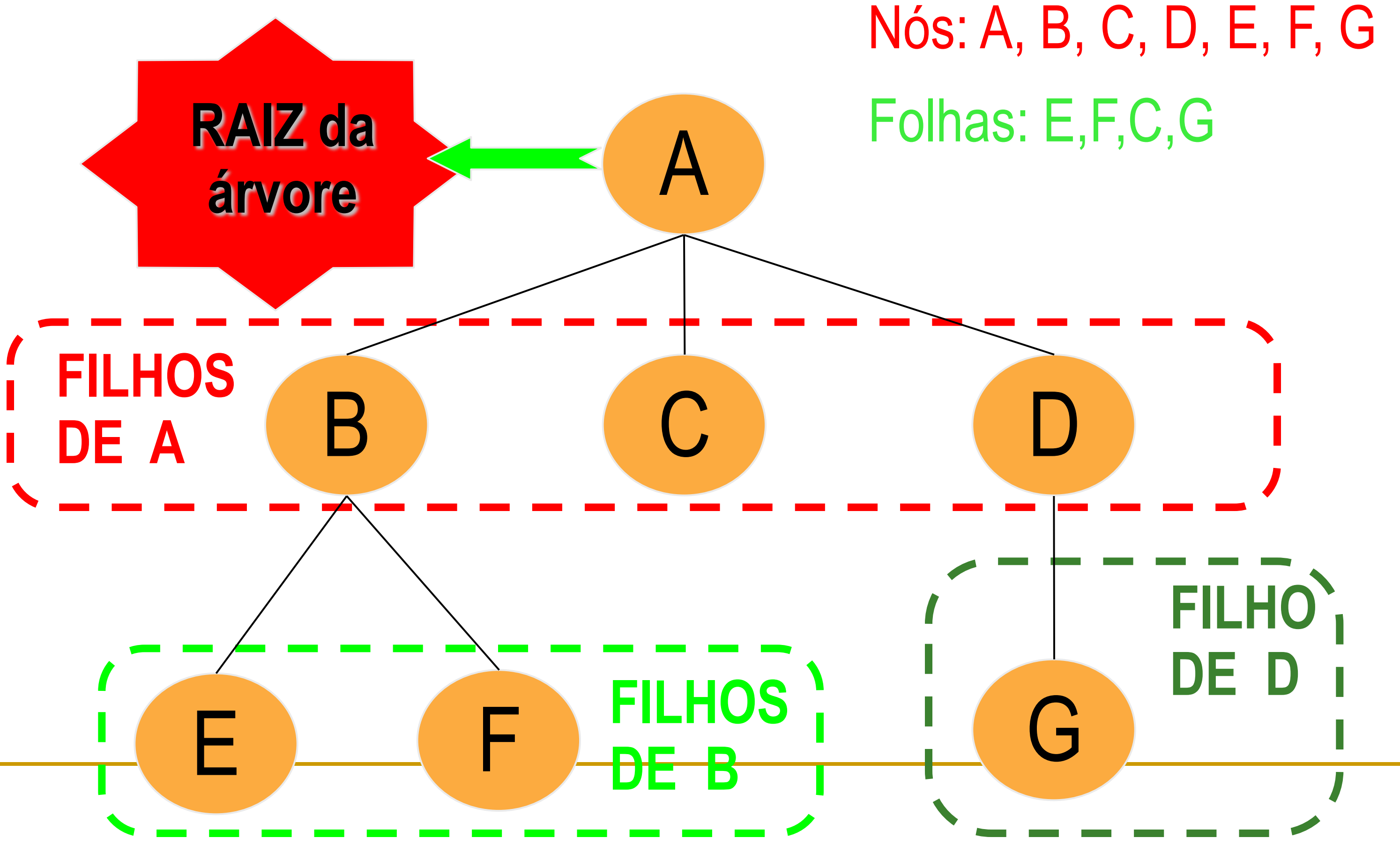
http://evolution.berkeley.edu/evolibrary/article/phylogenetics_02





Nós pais e filhos

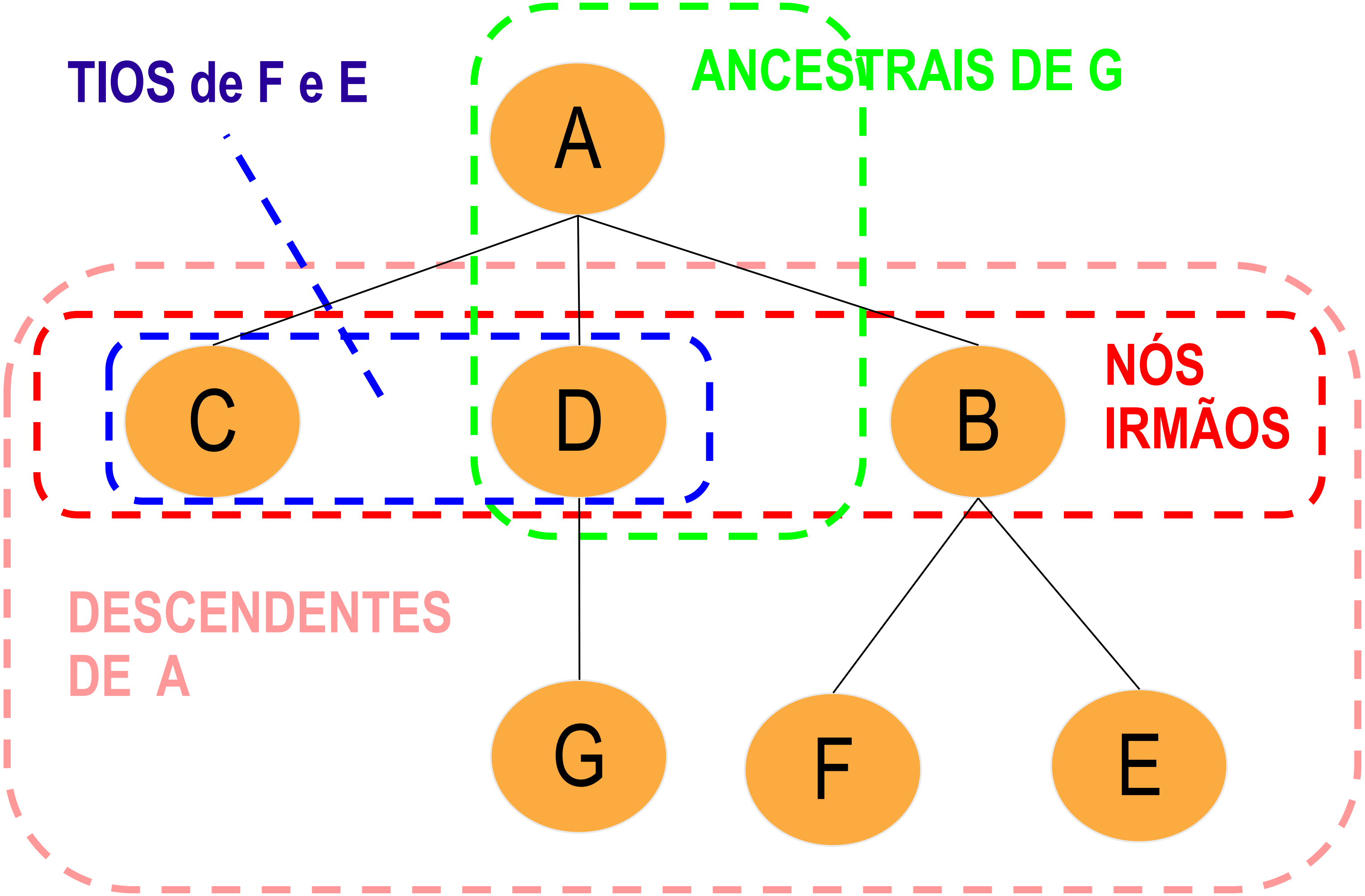
- Se um nó X é raiz de uma árvore, e um nó Y é raiz de uma sub-árvore de X, então X é **PAI** de Y e Y é **FILHO** de X



Ancentral e Descendente; Irmãos e Tios

- O nó X é um ANCESTRAL do nó Y (e Y é DESCENDENTE de X)
  - se X é o PAI de Y, ou se X é PAI de algum ANCESTRAL de Y
- Dois nós são IRMÃOS se são filhos do mesmo pai
- Se os nós Y_1, Y_2, ...Y_j são irmãos, e o nó Z é filho de Y_1, então Y_2,...Y_j são TIOs de Z

Exemplos


TIOS de F e E
ANCESTRAIS DE G
A
C
D
B
NÓS IRMÃOS
DESCENDENTES DE A
G
F
E

Conceitos

- Nível
  - não há definição única para o valor do nível da raiz
- Grau
- Caminho e comprimento do caminho
- Altura ou profundidade
- Árvore Ordenada
- Árvore Orientada (há autores que definem orientada como ordenada; não distinguem os 2 conceitos)
- Floresta
- Árvore Cheia

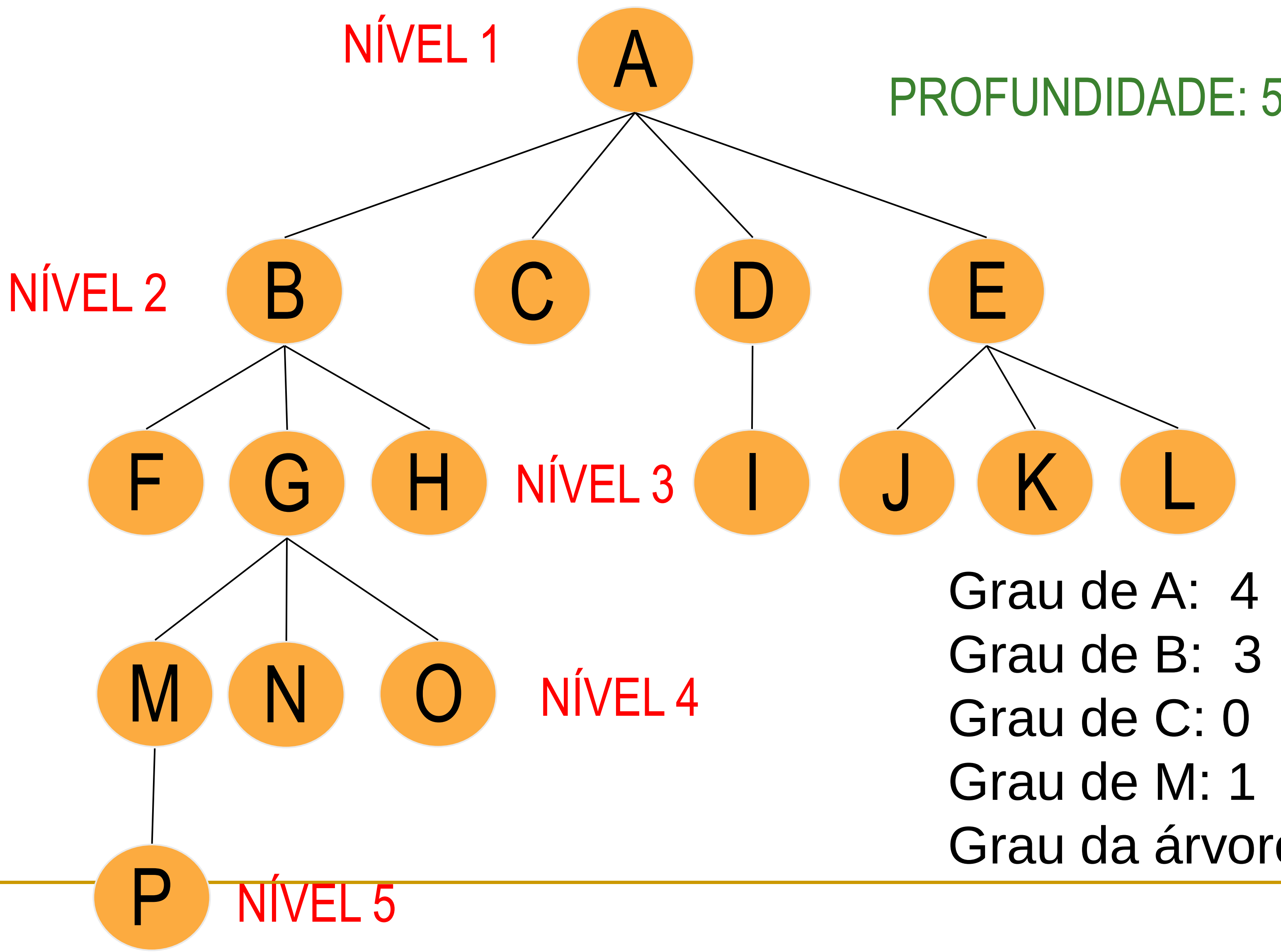
Conceitos

- O NÍVEL de um nó **X** é definido como:
  - O nível do nó raiz é 1
    - (esta definição não é universal – o nível da raíz pode ser 0)
  - O nível de um nó não-raiz é dado por (nível de seu nó PAI + 1)
- Os nós de maior nível são também nós-folha.

Conceitos

- O GRAU de um nó **X** pertencente a uma árvore é igual ao número de filhos do nó X
- O GRAU de uma árvore T é o maior entre os graus de todos os seus nós

Conceitos



Grau de A: 4
Grau de B: 3
Grau de C: 0
Grau de M: 1
Grau da árvore:4

Conceitos

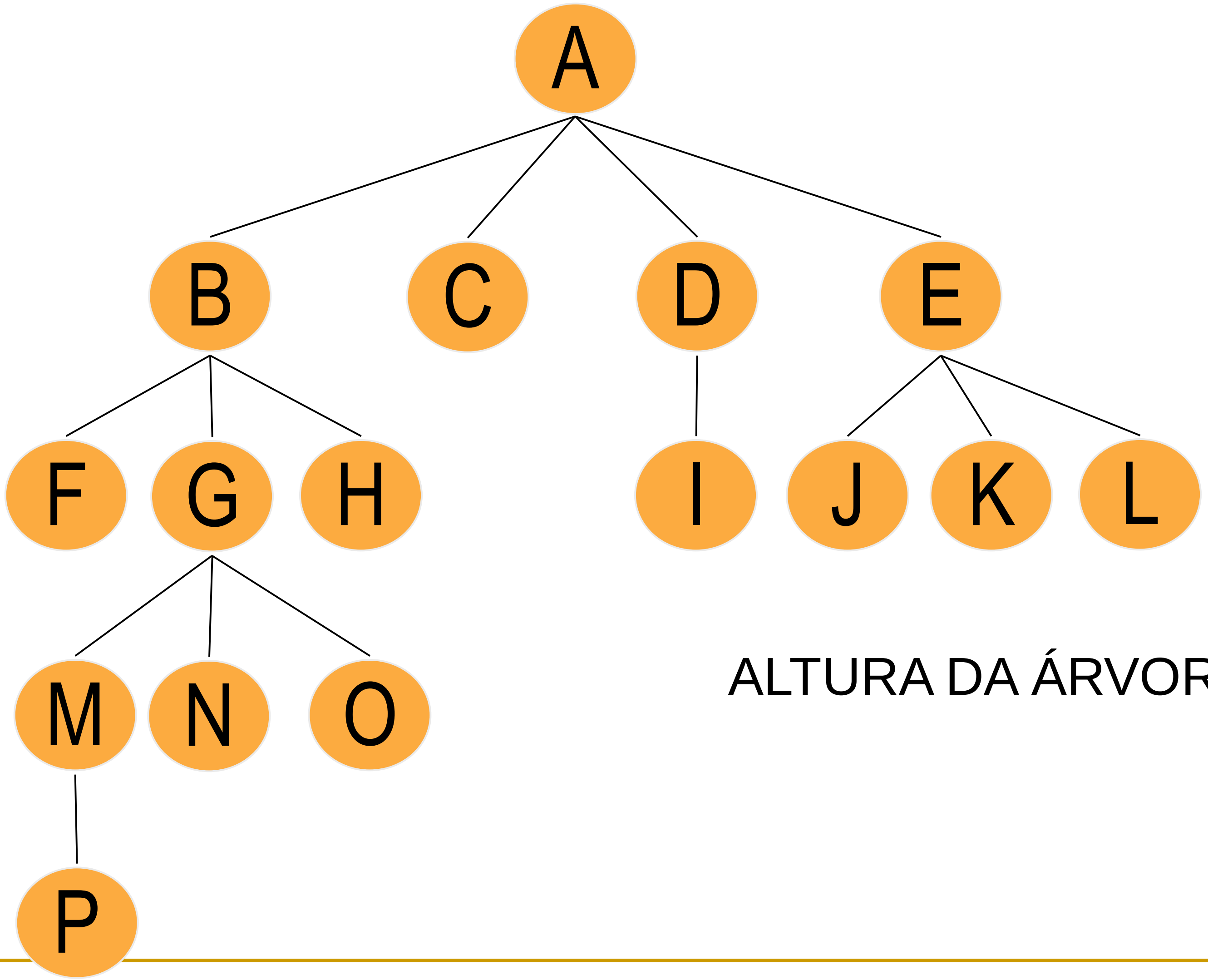
- Uma sequência de nós distintos v_1, ….v_k tal que cada nó v_{i+1} é filho de v_i é denominada um **CAMINHO** na árvore (diz-se que v_i alcança v_k).

- O número de arestas de um caminho define o **COMPRIMENTO** DO CAMINHO.

- A **ALTURA** ou **PROFUNDIDADE** de uma árvore **X** é dada pelo MAIOR NÍVEL de seus nós.
  - Aternativamente, corresponde ao número de nós do maior caminho entre a raiz e os nós folhas.

- Denota-se a altura de uma árvore com raiz **X** por **h(X)**, e a altura de uma sub-árvore com raiz **y** por **h(y)**

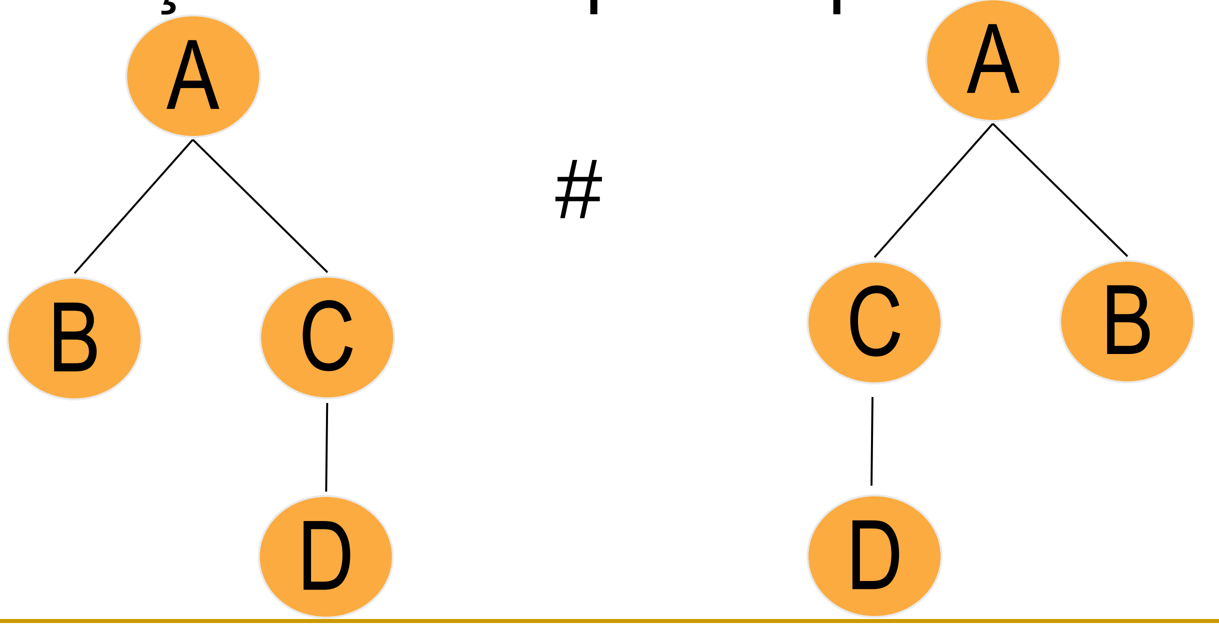
Conceitos



ALTURA DA ÁRVORE: 5

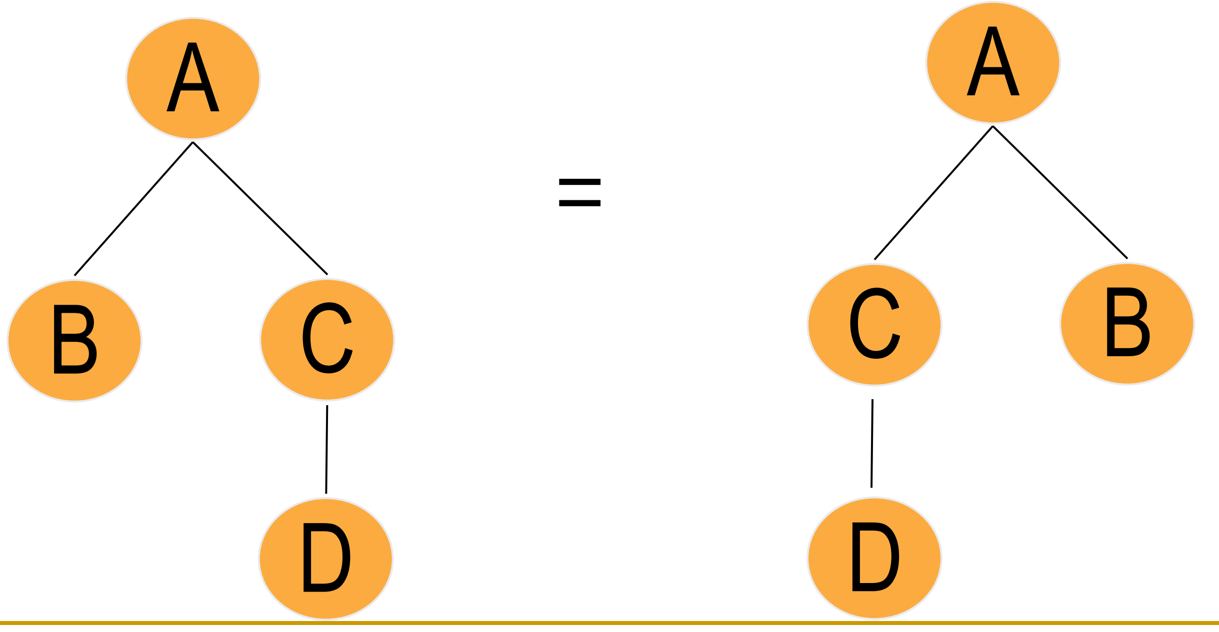
Conceitos

- Uma árvore é **ORDENADA** se considerarmos o conjunto de sub-árvores T_1, T_2, …T_n como um conjunto ordenado. Ordenação é da esquerda para direita.



Conceitos

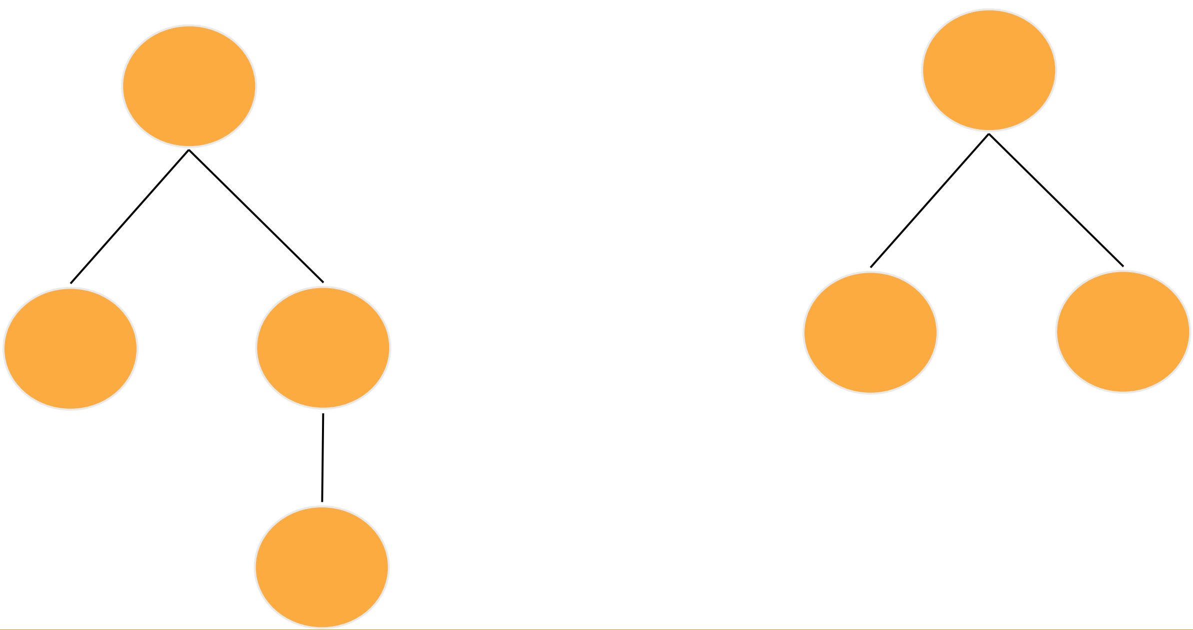
- Uma árvore é **ORIENTADA** se apenas a orientação relativa dos nós – e não sua ordem – está sendo considerada.



Vejam que aqui o C ainda é pai de D

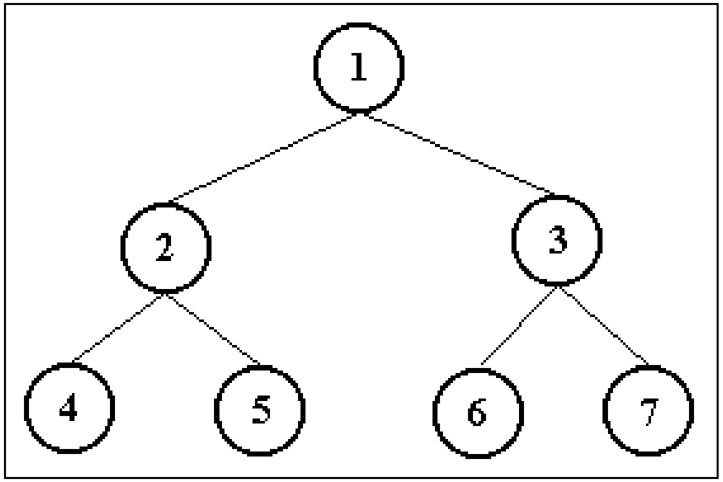
Conceitos

- Uma **FLORESTA** é um conjunto de 0 ou mais árvores distintas



Conceitos

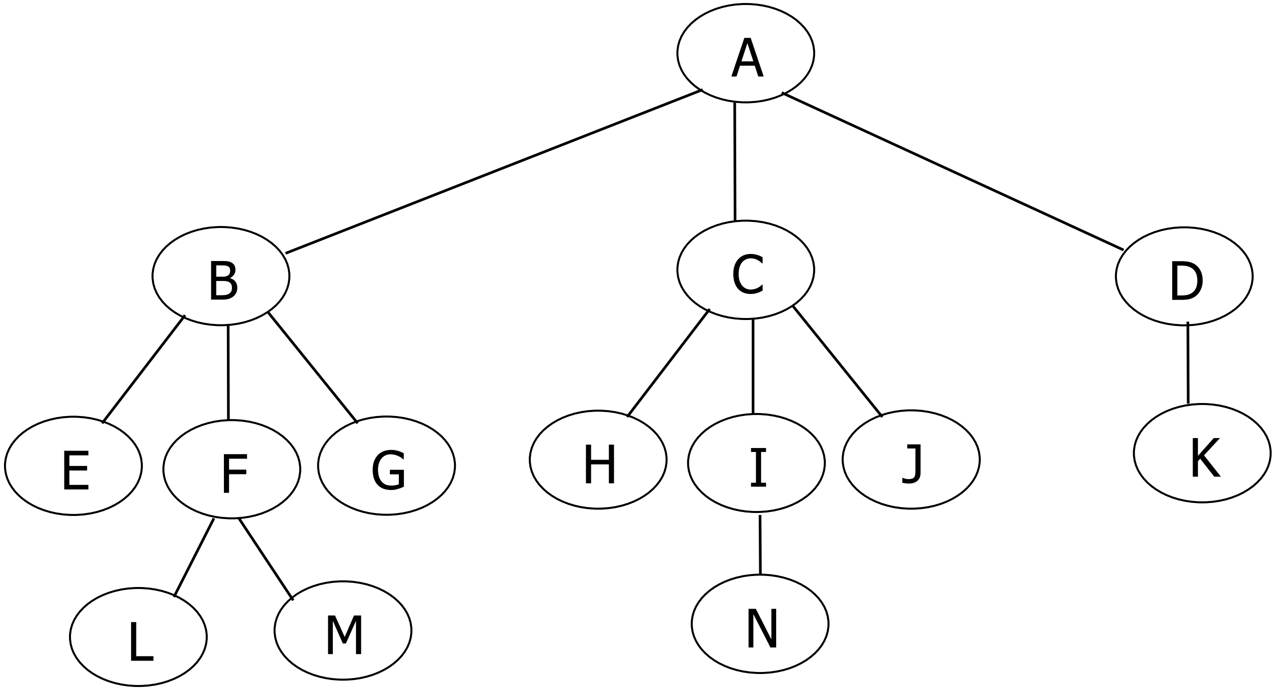
- Árvore cheia
  - Uma árvore de grau d é uma árvore cheia
    - se possui o número máximo de nós, isto é, todos os nós tem número máximo de filhos (exceto as folhas, logicamente) e todas as folhas estão na mesma altura



Exemplo de árvore cheia de grau 2

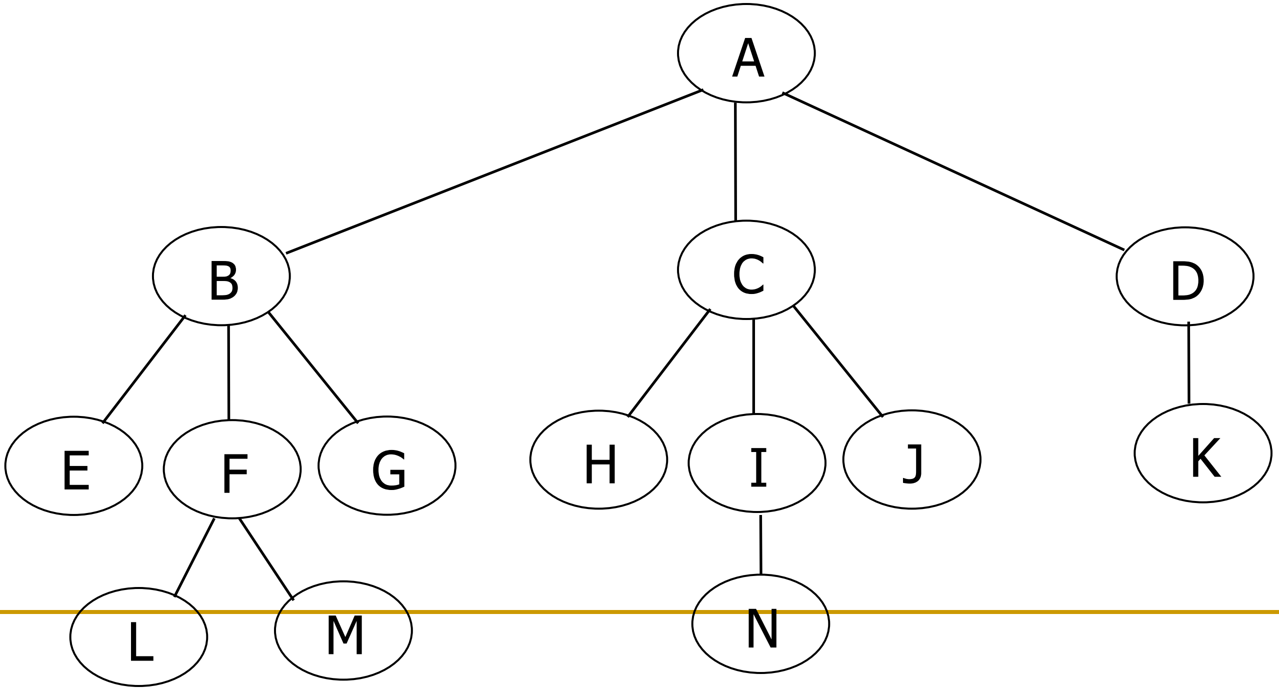
Árvores

- Considere a árvore abaixo
  - Quantas subárvores A tem?



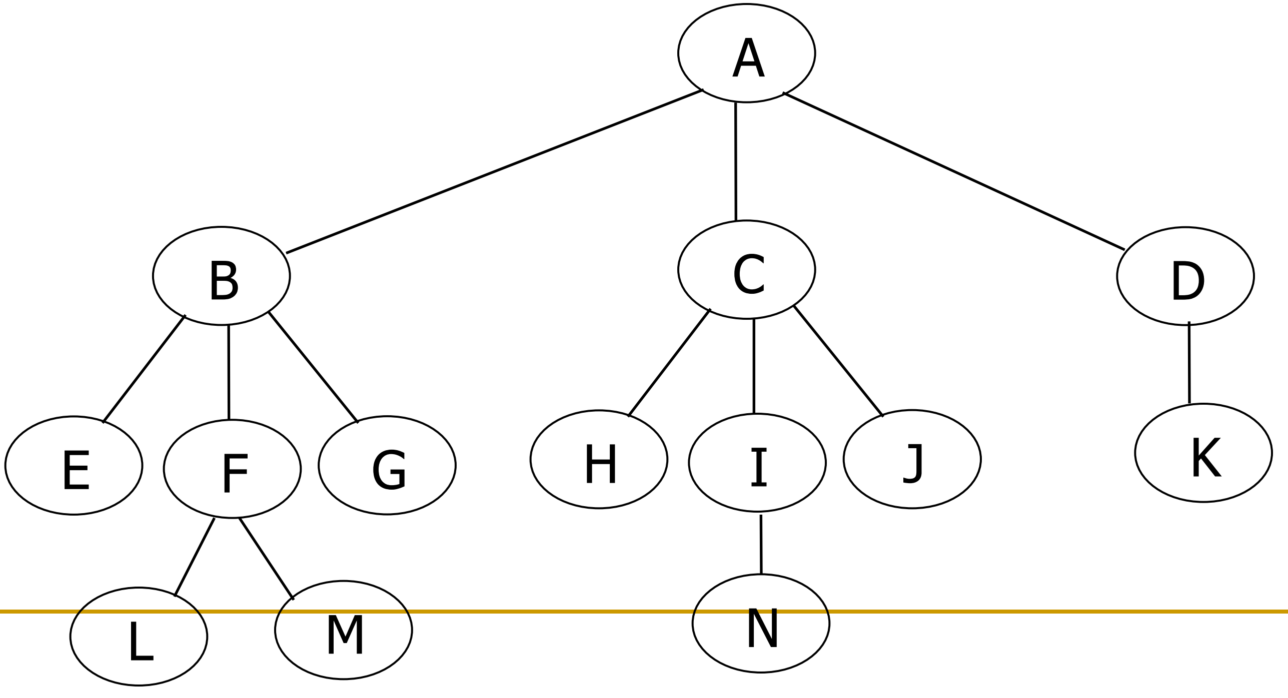
Árvores

- Considere a árvore abaixo
  - Quem são os filhos de A? E os descendentes de A?



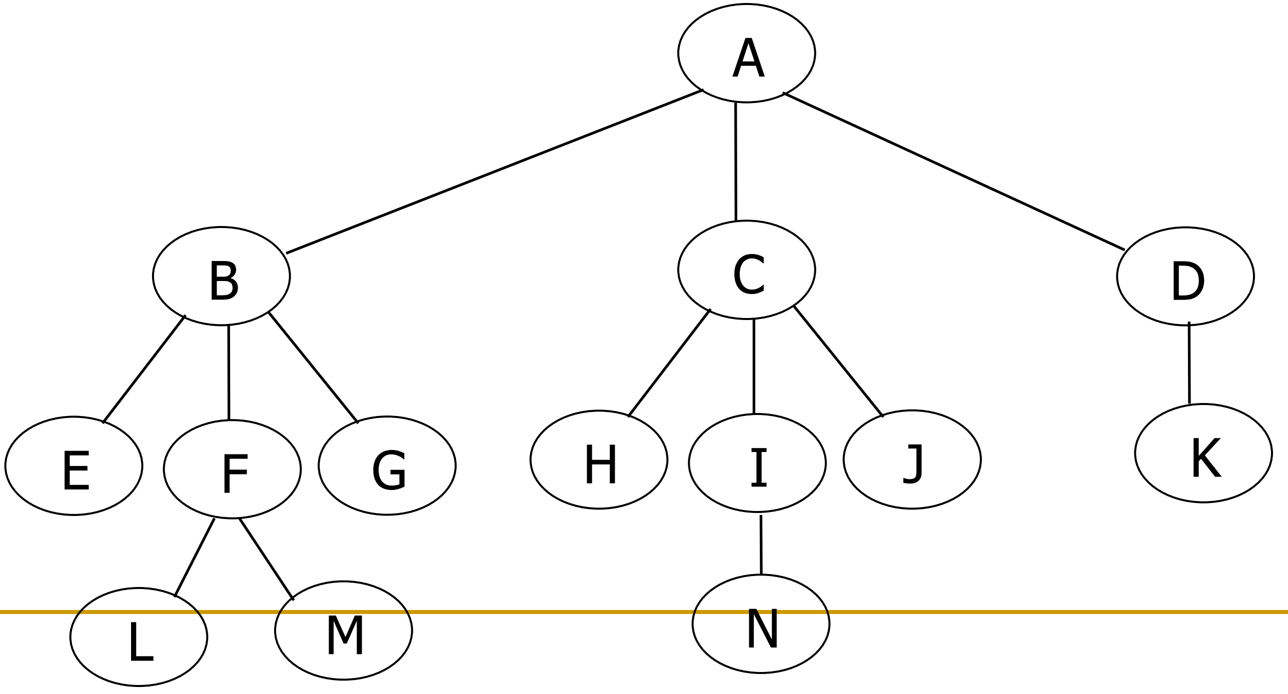
Árvores

- Considere a árvore abaixo
  - Quais são os nós folha dessa árvore?



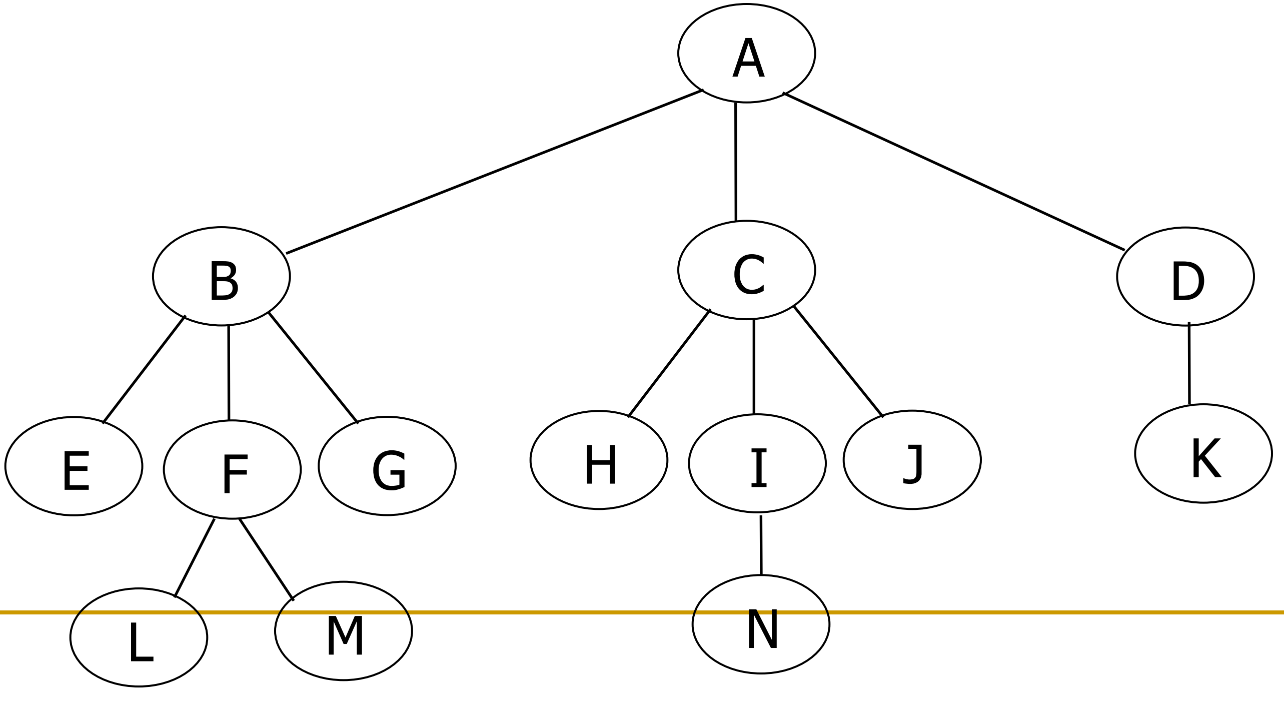
Árvores

- Considere a árvore abaixo
  - Qual o grau dessa árvore?

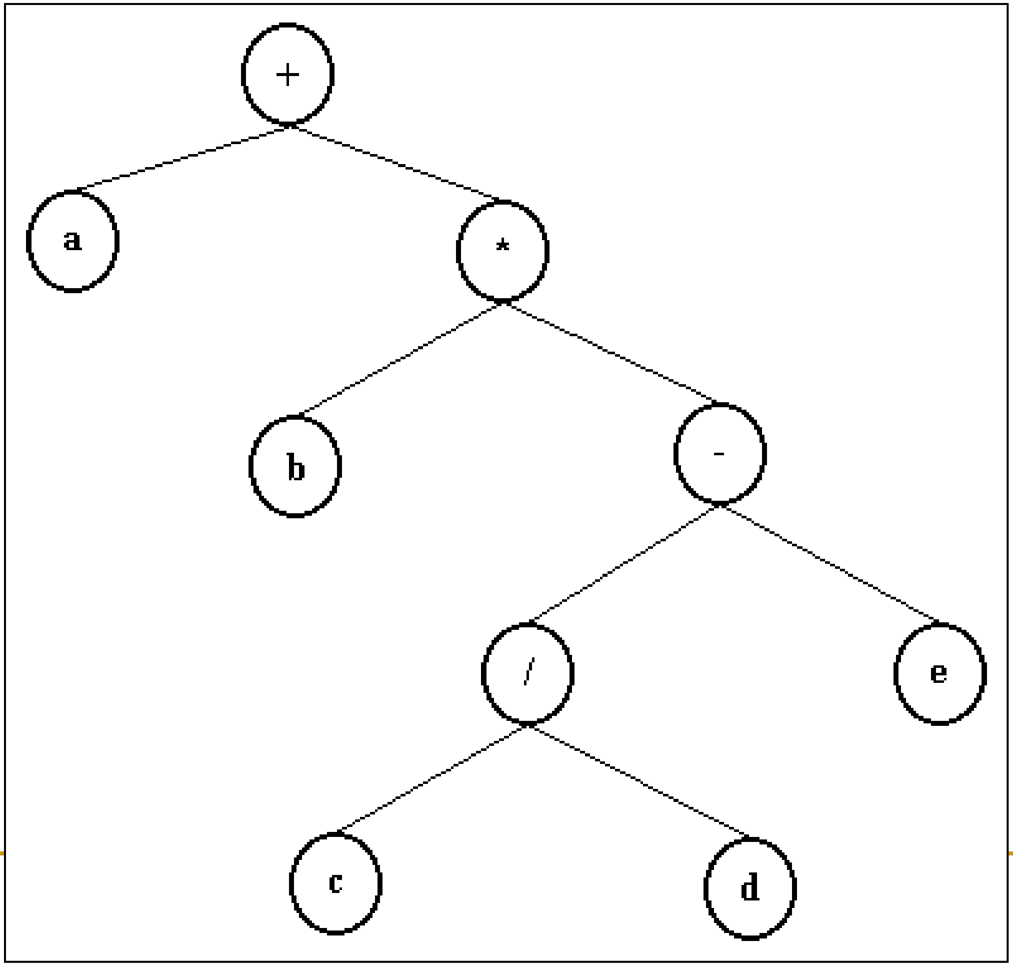


Árvores

- Considere a árvore abaixo
  - Qual a altura dessa árvore?



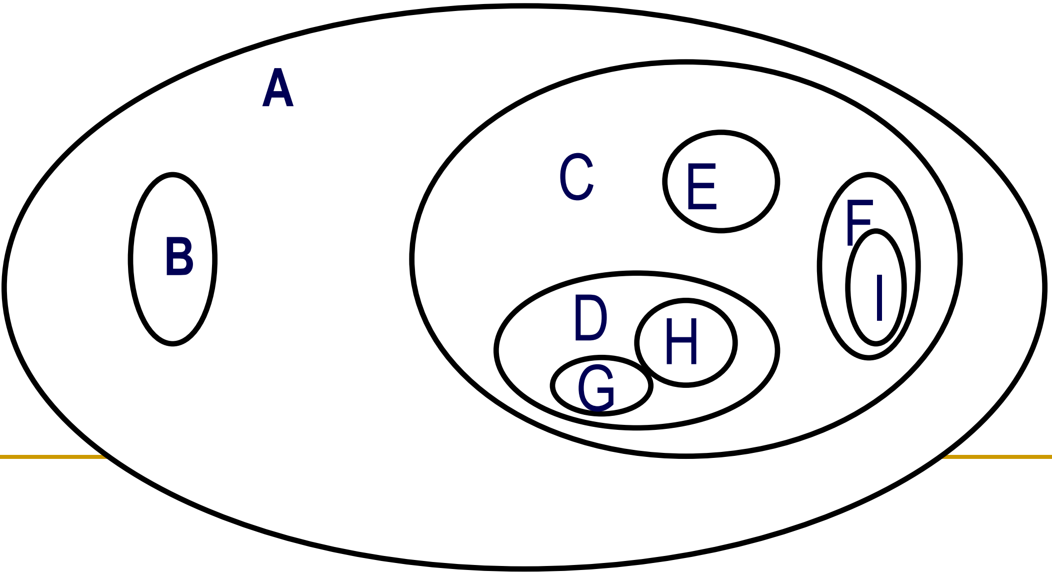
Exemplo: Representação da expressão aritmética com operadores binários (a + (b * ((c / d) - e)))



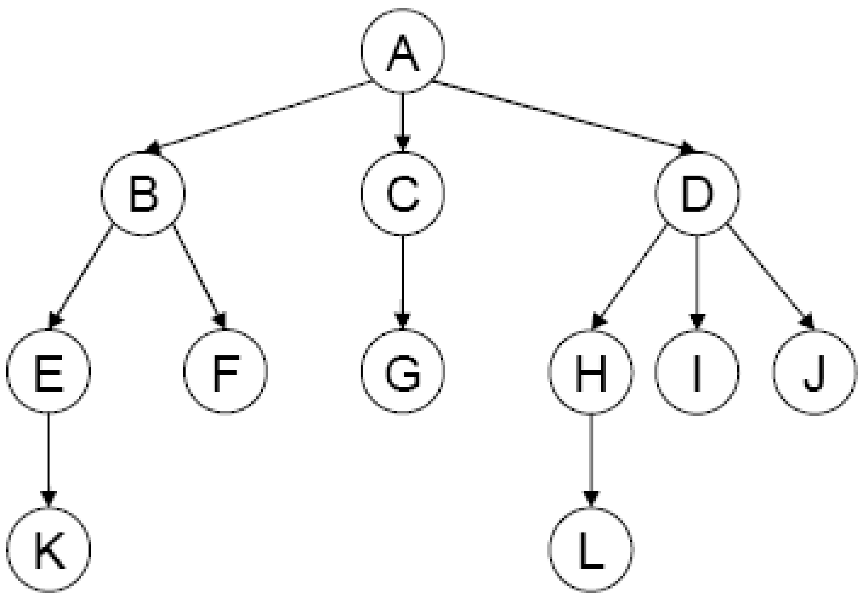
Cada operador é um nó da árvore e seus dois operandos representados como subárvores

Outras Representações Gráficas

- Representação por paragrafação
- Representação por parênteses aninhados
  - (A (B) (C (D (G) (H)) (E) (F (I))))
  - ou seja, uma lista generalizada!!
- Representação por Diagramas de Venn

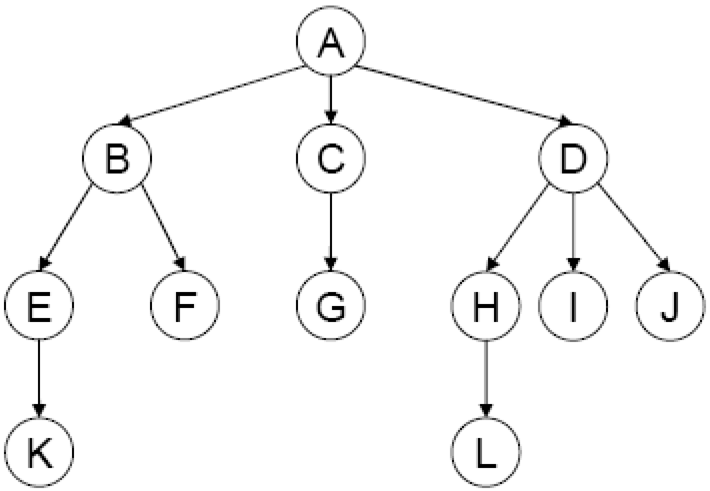


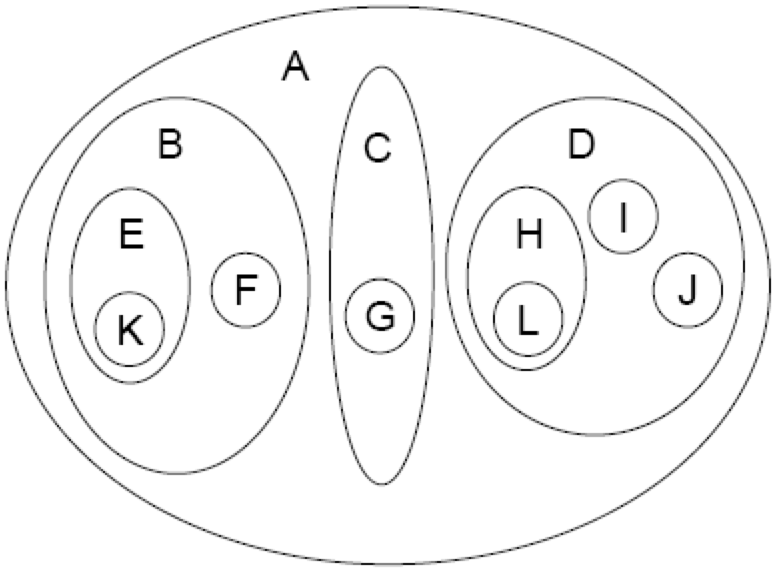
Representação por paragrafação



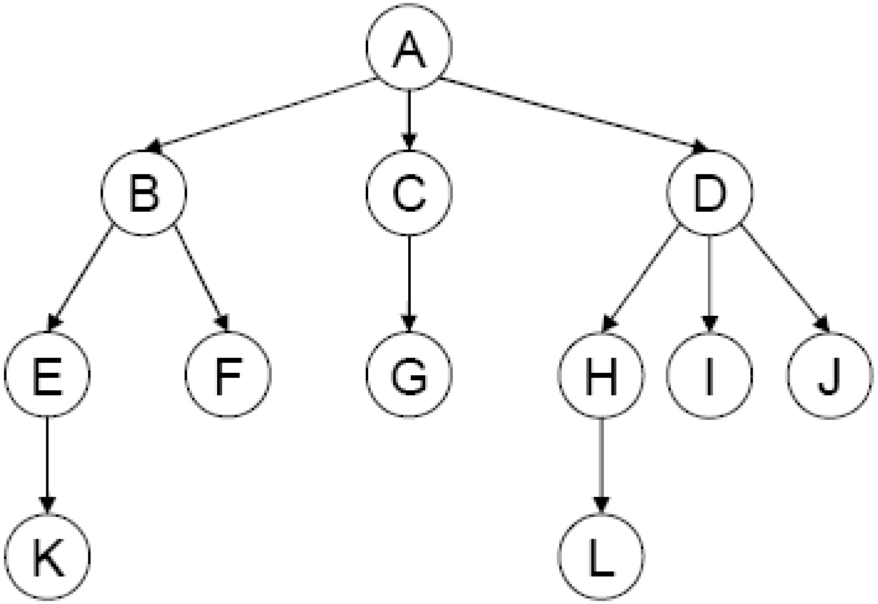
A
..B
....E
......K
....F
..C
....G
..D
....H
......L
....I
....J

Representação por Diagramas de Venn





Representação por parênteses aninhados



- ((A (B (E (K) (F)) C (G) D (H (L) (I) (J))))

Implementação de Árvores

- Para implementar em C qualquer tipo de árvore é necessário modelar as relações hierárquicas entre os valores dos nós.
  - A forma mais fácil de se representar as relações pai/filho é incluir um ponteiro no pai, apontando para o filho.
- Definimos um nó como uma estrutura e a árvore é um ponteiro para a estrutura.
- A definição é recursiva:
  - Árvores são ponteiros para nós
  - Nós são estruturas que contém árvores

Árvores Genealógicas

- Cada nó consiste de um nome

de uma pessoa e um conjunto de

ponteiros para seus filhos.

- Armazenando os ponteiros para os filhos em um vetor, o nó seria:

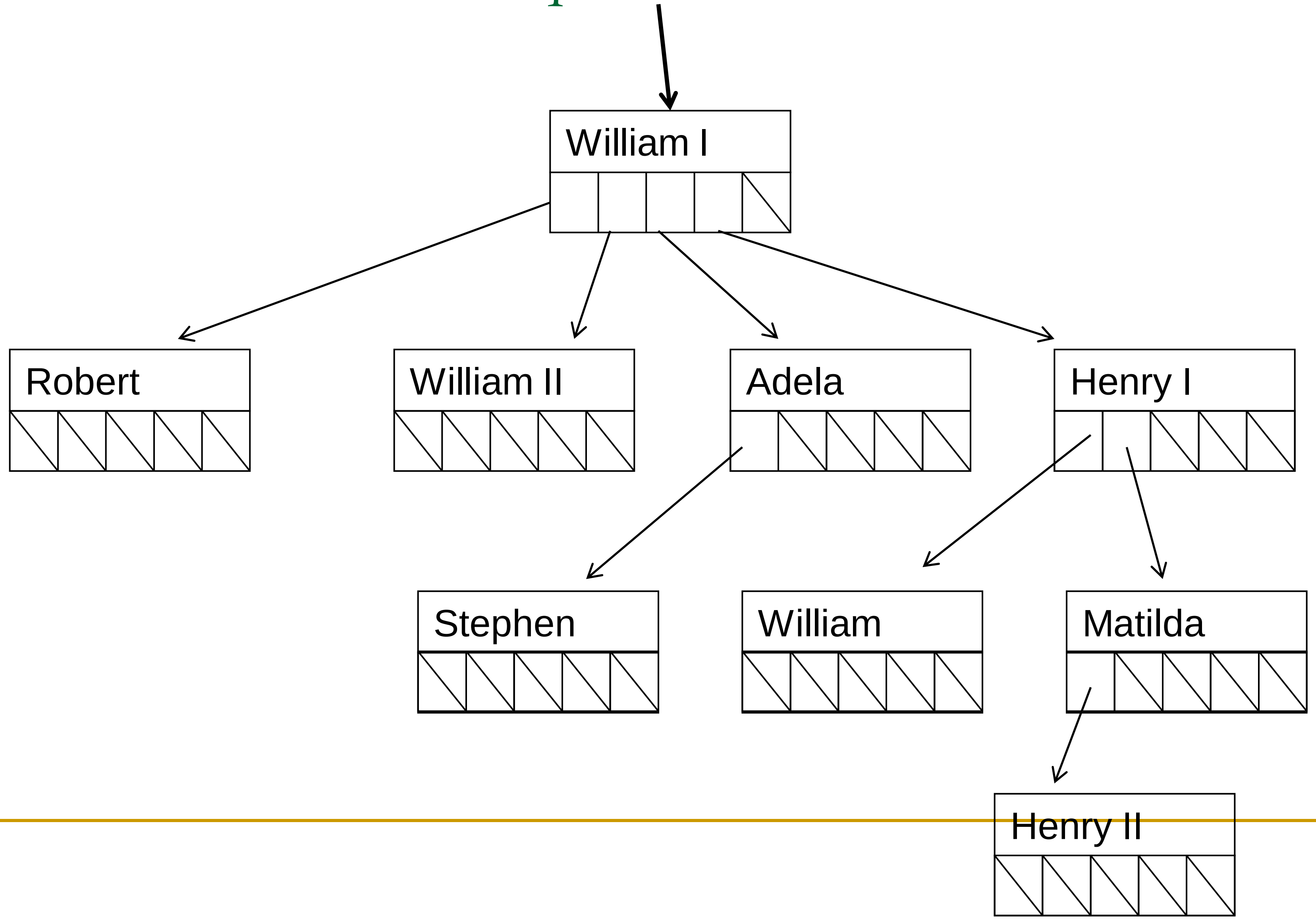
```
struct no_familia {
    char nome[10];
    struct no_familia* filhos[MAX_Filhos];
}
```

Tree.c

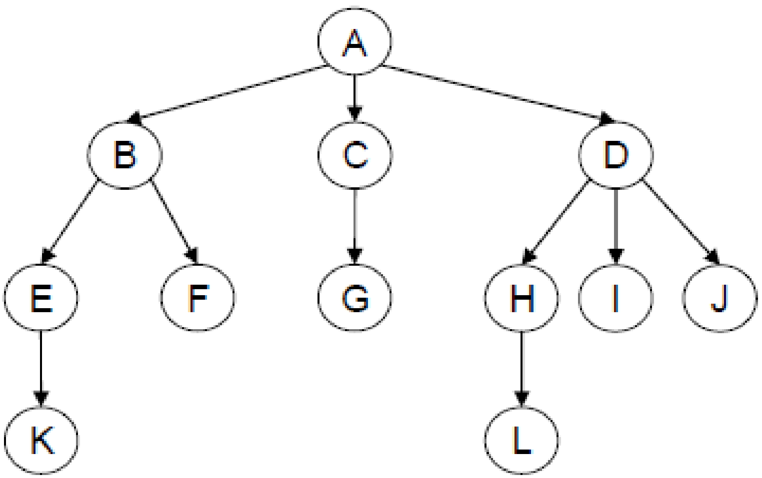
```
typedef struct no_familia* root;
```

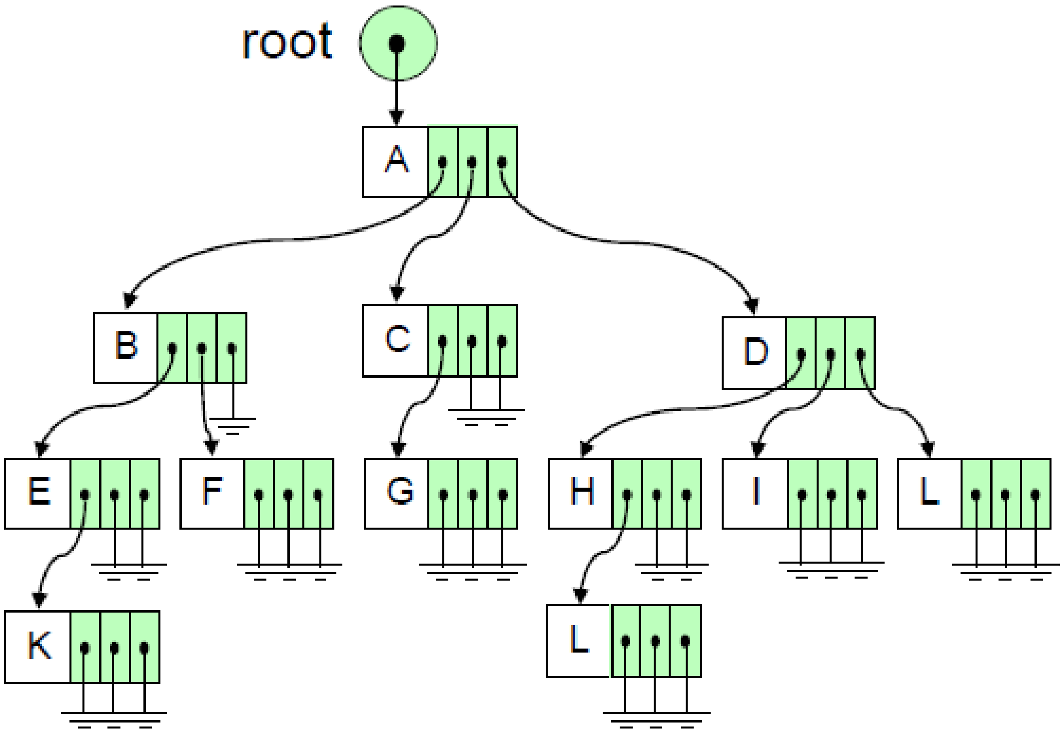
Tree.h

Família real que governou a Inglaterra após 1066 quando Willian o conquistador se tornou Rei Willian I



Árvores Ternárias - exemplo



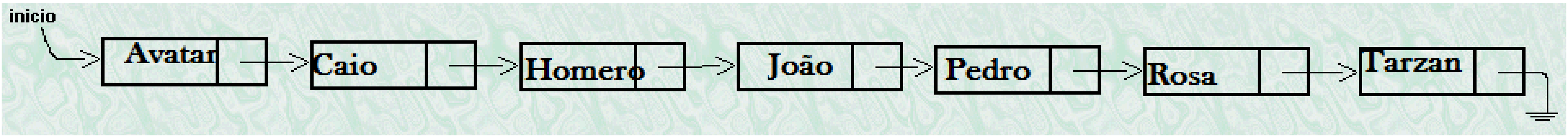


A motivação para árvores binárias ...

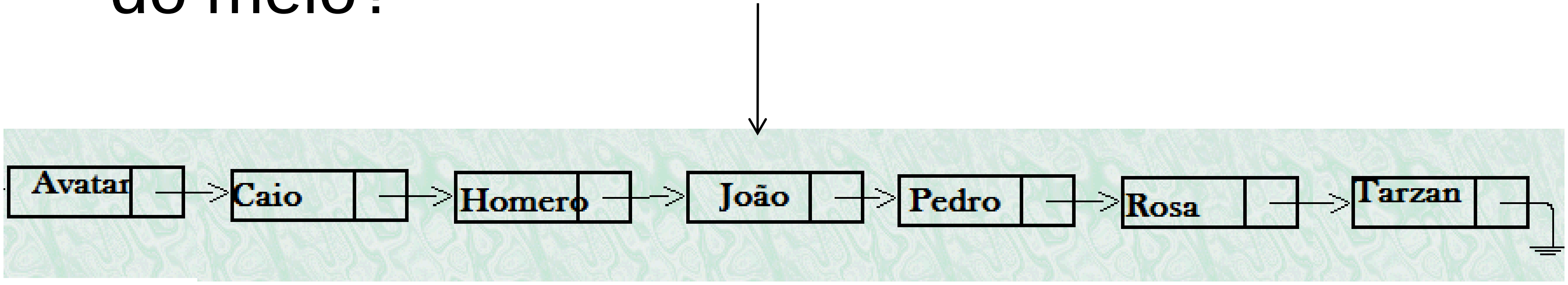
- Embora seja possível fazer um TAD para árvores de famílias, é mais efetivo fazer em um ambiente mais simples.
- Em muitos contextos de programação é razoável restringir o número de filhos para facilitar a programação.

- Uma das subclasses mais importantes de árvores, com muitas aplicações práticas, são as árvores binárias (AB), que serão vistas agora...
  - Imaginem uma lista de 7 nomes, ordenados lexicograficamente....

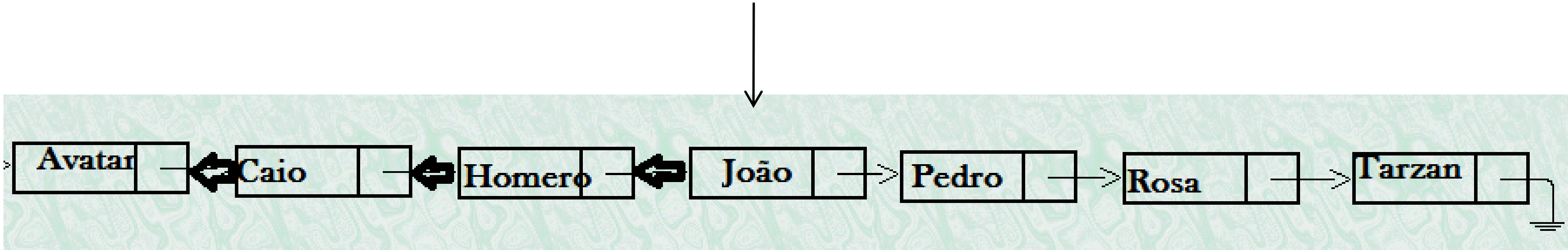
Lista ligada com 7 nomes, ordenados lexicograficamente



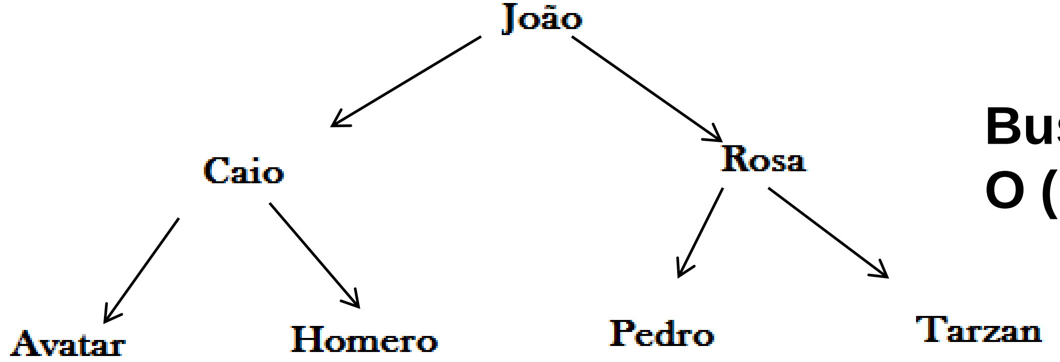
- Fácil buscar o primeiro, mas deve percorrer a lista para achar o elemento do meio: O(N).
- E se houvesse um ponteiro para o elemento do meio?



- Fácil buscar o do meio, mas perdemos o acesso aos 3 primeiros.
- E se invertêssemos os ponteiros à esquerda?
- Toda a lista ficaria acessível



- Se aplicarmos este algoritmo recursivamente acabaríamos com a seguinte AB de Busca. Os nós a esquerda são menores que a raiz e os da direita são maiores que a raiz.



Busca:
O (log N)