Árvore

Motivação

Algumas questões sobre as estruturas vistas até aqui:

- vetor ordenado – problemas na inserção e na remoção de um elemento
- lista – pesquisa de um elemento
- pilha, fila, lista – linear

Árvore – estrutura de dados não linear, hierárquica

Representação

- grafos
- conjuntos aninhados (diagrama de inclusão)
- parênteses aninhados
- paragrafação (indentation)

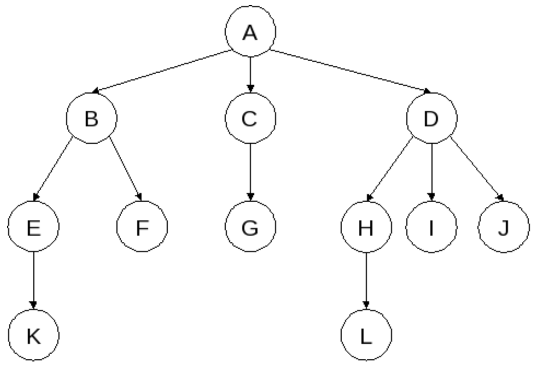
Árvore - definição

Definição recursiva (elegante e eficaz) – uma árvore T é um conjunto de nós tal que:

(a) T é vazia ou

(b) T consiste de um nó raiz e o restante dos nós pode ser particionado em $m \geq 0$ conjuntos disjuntos de árvores do mesmo tipo T denominadas subárvores.

Árvore - grafo

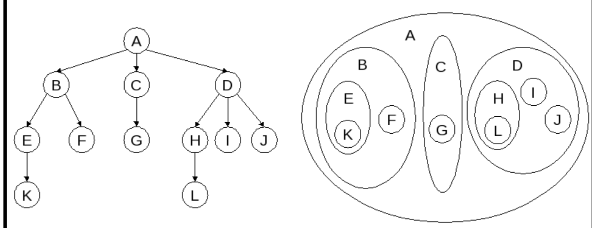


Árvore - outra definição

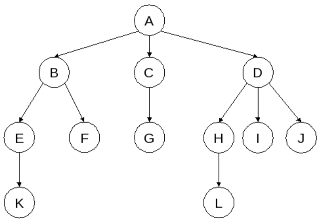
Conjunto finito de nós, com informação e que têm uma relação entre si do tipo pai-filho. Se a árvore não é vazia (tem nós):

- existe um nó especial – a raiz – que não tem pai
- todo o nó da árvore (excepto a raiz) tem um único pai
- um nó pode ter 0, 1 ou mais filhos

Árvore - conjuntos aninhados



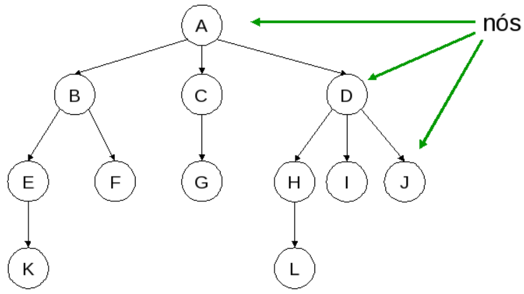
Árvore - parênteses aninhados



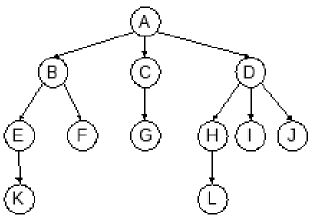

```
(A (B (E(K))
      (F))
   (C(G))
   (D(H(L))
     (I)
     (J)))
```

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

Nós

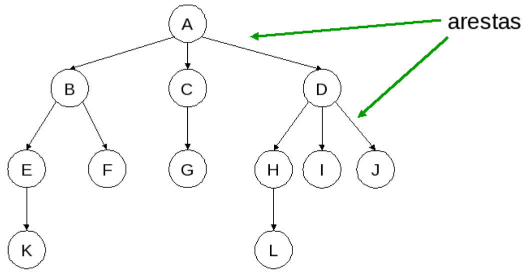


Árvore - paragrafação




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

Arestas (Arcos)

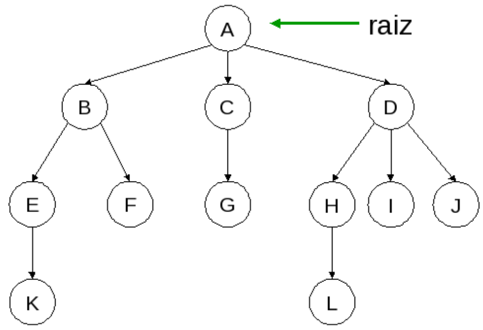


Exercício 1

Use as diversas representações para a árvore:

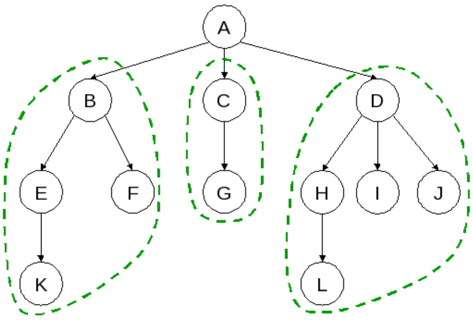
- A é raiz
- B, C e D são filhos de A
- E tem como pai B
- C é pai de F e G
- H é filho de D
- I é filho de H
- J e K são irmãos de I

Raiz



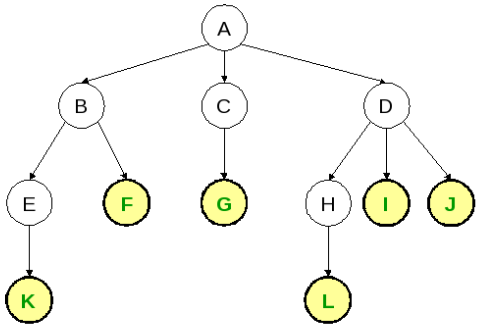
Subárvores

Nó A tem 3 subárvores cujas raízes são B, C e D:



Nó folha

Nó sem filhos

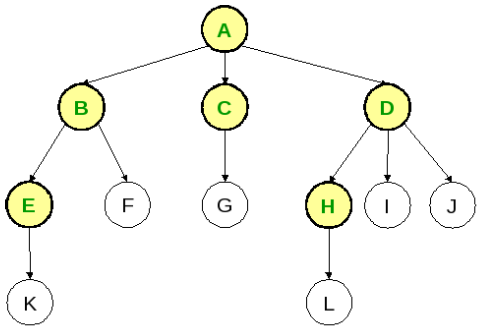


Subárvores

Quais são as subárvores de B?

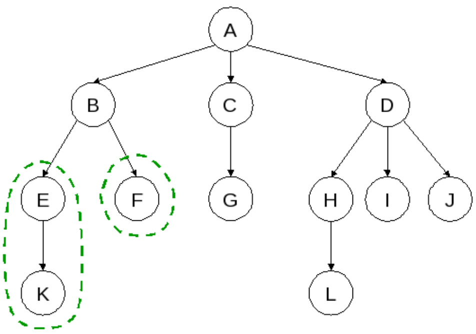
Não-folha, interior ou interno

Nó com pelo menos um filho



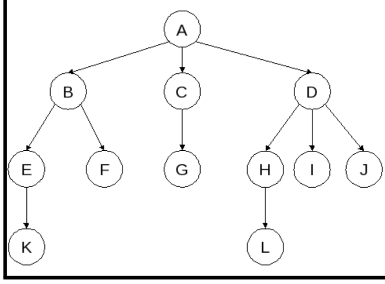
Subárvores

Subárvores de B:



Pai

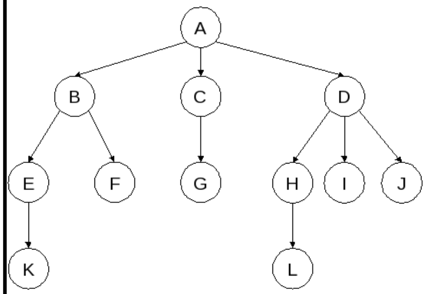
Raízes das subárvores de X são os filhos de X



Nó Pai	Nós Filhos
A	B, C, D
B	E, F
C	G
D	H, I, J
E	K
F	-
G	-
H	L
I	-
J	-
K	-
L	-

Irmãos

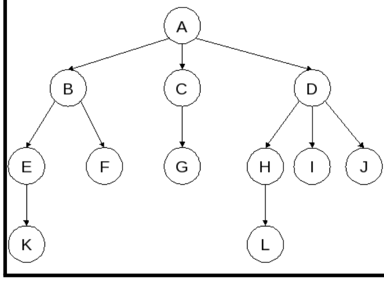
Possuem o mesmo pai



Irmãos
B, C, D
E, F
H, I, J

Grau de um nó

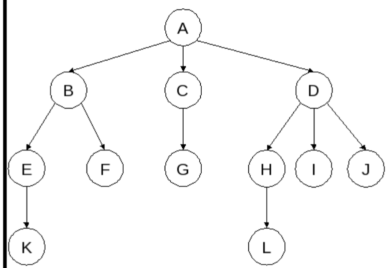
Número de filhos de um nó



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

Avô e demais parentes

Possuem o mesmo pai

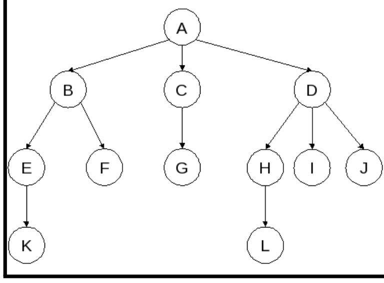


Nós	Avô
E,F,G,H,I,J	A
K	B
L	D

Nós	Bisavô
K, L	A

Grau de um nó

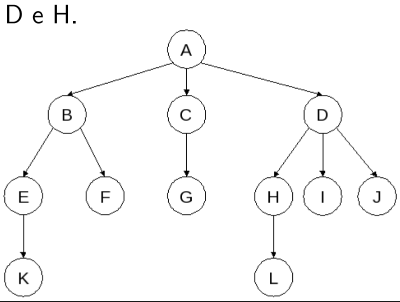
Grau do nó folha = 0



Nó	Grau
A	3
B	2
C	1
D	3
E	1
F	0
G	0
H	1
I	0
J	0
K	0
L	0

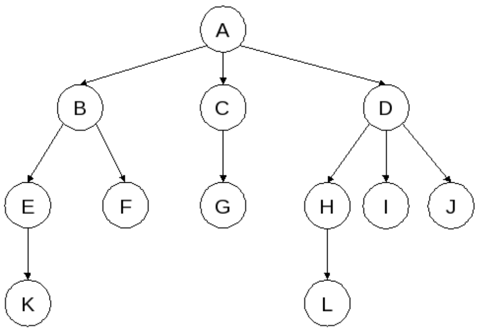
Antecessores (ancestral)

Todos os nós no caminho entre a raiz e o respectivo nó. No exemplo, os antecessores de L são A, D e H.



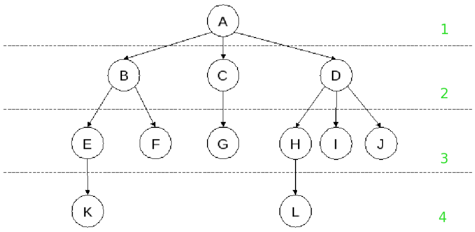
Grau da árvore

Grau máximo



Nível (ou profundidade)

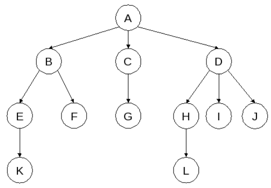
- raiz está no nível 1
- considerando um nó no nível i, seus filhos estarão no nível $i+1$



Alguns autores consideram a raiz no nível 0.

Altura

- número de nós do caminho da raiz até o nó.
- considerando um nó no nível i, seus filhos estarão no nível $i+1$



Nós	Altura
K, F, G, L, I, J	1
E, C e H	2
B e D	3
A (árvore)	4

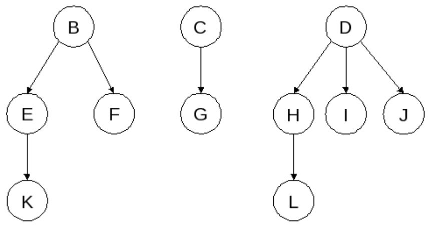
A altura da árvore nula é definida como 0.

Caminho entre v_1 e v_k

- sequência de nós distintos $v_1, v_2, ..., v_k$ tal que sempre existe a relação "v_i é filho de v_{i+1}" ou "v_i é pai de v_{i+1}", $1 \leq i < k$
- v_1 alcança v_k ou v_k é alcançado por v_1
- formado pela sequência de $k-1$ pares de nós $(v_1, v_2), (v_2, v_3), ..., (v_{k-1}, v_k)$, onde cada par $(v_i,\ v_{i+1})$ é uma aresta ou arco, $1 \leq i < k$
- formado por k vértices e tem comprimento $k-1$

Floresta

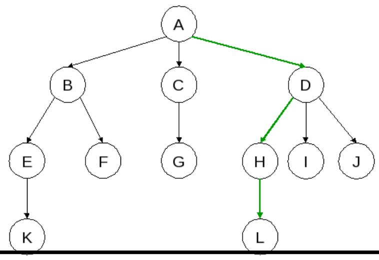
Conjunto de zero ou mais árvores



3 árvores que compõem uma floresta

Caminho

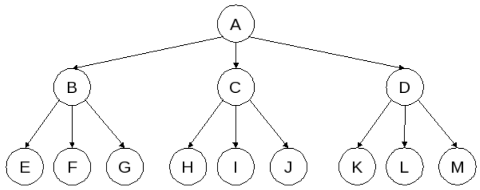
- A, D, H, L é um caminho entre A e L, formando pela sequência (A,D), (D,H), (H,L)
- o comprimento do caminho entre A e L é 3



Árvore completa

Uma árvore de grau d é completa (cheia) se:

- todos os nós tem exatamente d filhos, exceto as folhas e
- todas as folhas estão na mesma altura



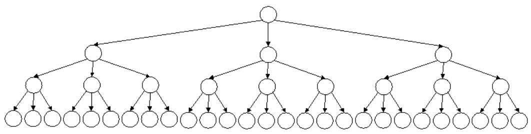
Número Máximo de Nós

O número máximo de nós n(h,d) em uma árvore de altura h é atingido quando todos os nós possuírem grau d, exceto os de nível h, que não possuem subárvores

- Nível 1 contém d^0 (um) nó (raiz)
- Nível 2 contém d^1 descendentes da raiz
- Nível 3 contém d^2 descendentes
- Nível i contém d^{i-1} descendentes

Número Máximo de Nós

- Árvore de grau 3 e altura 4
- nível 1 = 1, nível 2 = 3, nível 3 = 9, nível 4 = 27
- n(4,3) = 1 + 3 + 9 + 27 = 40 nós



Número Máximo de Nós

Portanto, o número máximo de nós n=n(h,d) é soma do número de nós em cada nível, ou seja:

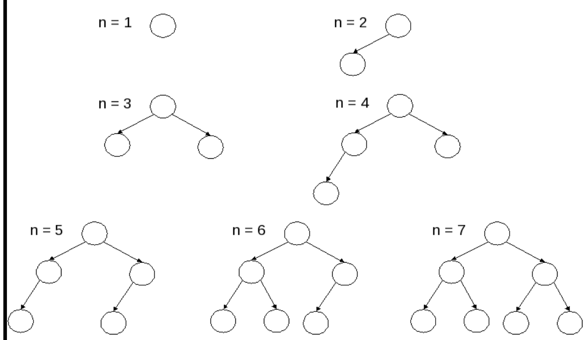
$$n = n(h, d) = \sum_{i=0}^{h-1} d^i = d^0 + d^1 + d^2 + ... + d^{h-1}$$

$$\sum_{i=0}^{h-1} d^i = \frac{d^h - 1}{d - 1}, d > 1$$

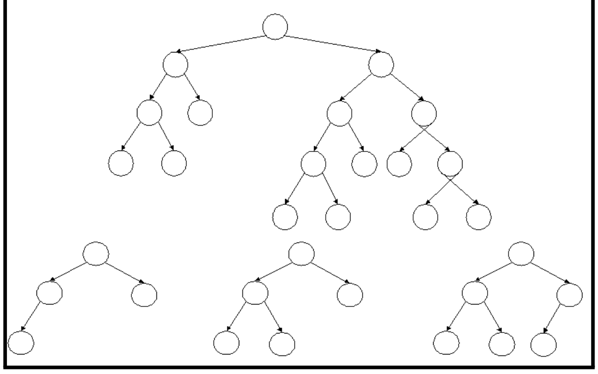
Árvores Balanceadas

- para cada nó, a altura de suas subárvores diferem, no máximo, de uma unidade
- perfeitamente balanceada – para cada nó, os números de nós em suas subárvores diferem, no máximo, de uma unidade
- todas as árvores perfeitamente balanceadas também são árvores balanceadas

Perfeitamente Balanceadas de grau 2

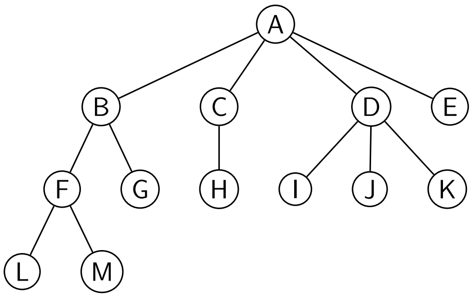


Balanceadas de grau 2

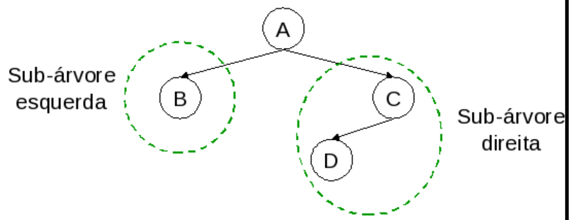


Exercício 2

Use as diversas representações para a árvore:



Exemplo

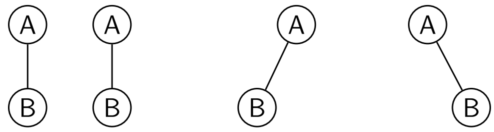


Exercício 3

Para as árvores dos dois exercícios identifique:

- as subárvores de cada nó
- todas as relações de parentesco entre os nós.
- os nós internos e nós folhas
- caminhos para os nós folhas
- altura da árvore e de cada nó
- grau da árvore e de cada nó
- balanceada ou perfeitamente balanceada?

Árvores Binárias



Árvores iguais Árvores Binárias Diferentes

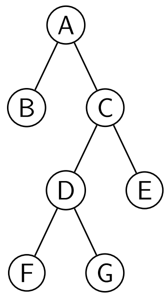
Árvore Binária

São estruturas do tipo árvore, onde o grau de cada nó é menor ou igual a 2

- pode ser vazia
- raiz + subárvore esquerda + subárvore direita
- as subárvores devem ser árvores binárias
- especificar subárvore esquerda ou direita caso o grau do nó seja 1

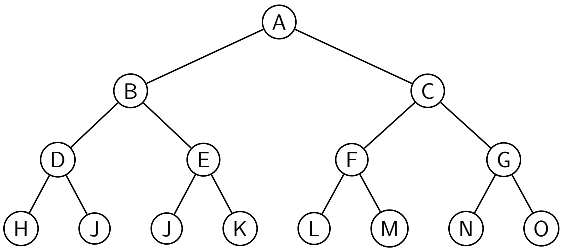
Árvore Estritamente Binária

Todo nó não-folha possui subárvore esquerda e direita não-vazias

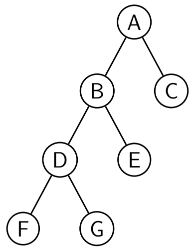


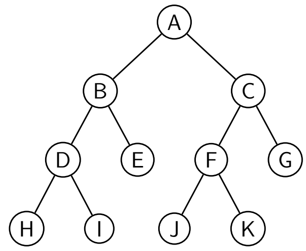
Árvore Binária Completa

Árvore estritamente binária em que todas as folhas estão no nível l



Estritamente Binária

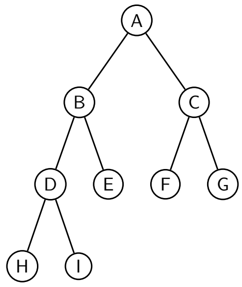


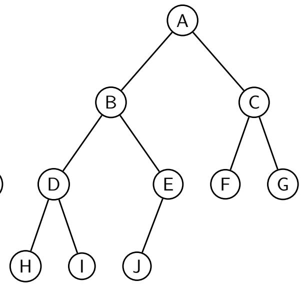


Número de Nós

- m nós no nível $l \Rightarrow 2m$ nós no nível $l+1$
- máximo de 2^{l-1} nós no nível l
- $\sum_{i=0}^{h-1} 2^i = 2^0 + 2^1 + 2^2 + ... + 2^{h-1} = 2^h - 1$
- árvore completa: 2^{h-1} folhas e $2^{h-1} - 1$ nós internos
- se n é o número total de nós em uma AB completa, sua pronfundidade é $\log_2(n+1)$

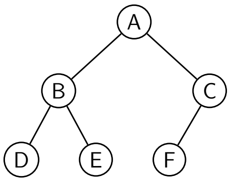
Quase Completa





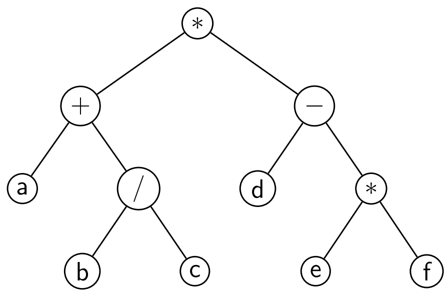
Árvore Binária Quase Completa

- cada folha está no nível l ou $l-1$
- para cada nó nl com descendente direito no nível l, todos os descendentes esquerdos de nl que forem folhas estão também no nível l.



Aplicação

Representação de Expressões aritméticas: (a+b/c)*(d-e*f)



Árvores de Expressão

- estritamente binária
- nós internos – operadores
- nós folhas – operandos
- raiz contém operador que deve ser aplicado aos resultados das avaliações representadas pela subárvore esquerda e direita

Árvores de Expressão

Desenhe a árvore das seguintes expressões:

- $A + B * C$
- $(A + B) * C$
- $A + (B - C) * D \wedge (E * F)$
- $(A + B * C) \wedge ((A + B) * C)$
- $A \wedge B * C - D + E/F/(G - H)$
- $((A + B) * C - (D - E)) \wedge (F - G)$

Percursos em AB

R – raiz, E e D – subárvores esquerda e direita, respectivamente

- Pré-ordem (pre-order): R, E, D – visitar a raiz antes das subárvores
- Em ordem (in-order): E, R, D
- Pós-ordem (post-order): E, D, R
- largura

Percurso Em Ordem

emOrdem(x)
Entrada: Subárvore com raiz x.

```
se x ≠ NULL faça
    emOrdem(x.esquerda)
    print x.chave
    emOrdem(x.direita)
```

Análise de complexidade

Se x é a raiz de uma subárvore de n nós, então a chamada `preOrdem` demora o tempo $\Theta(n)$.

$$T(n) = \begin{cases} c, & n = 0 \\ T(k) + T(n-k-1) + d, & n > 1, \end{cases}$$

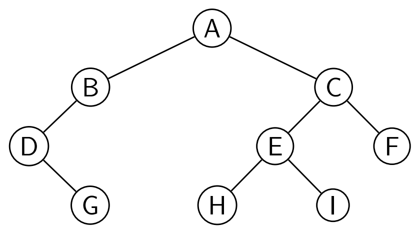
Pelo método da subtituição, provamos que
$T(n) \leq (c + d)n + c$.
Para $n = 0, (c + d)0 + c = c$

$$\begin{aligned} T(n) &\leq T(k) + T(n-k-1) + d \\ &= ((c+d)k + c) + ((c+d)(n-k-1) + c) + d \\ &= (c+d)n + c - (c+d) + c + d \\ &= (c+d)n + c \end{aligned}$$

A função **emOrdem** demora um tempo pequeno e constante em uma subárvore vazia.Então $T(0) = c$.

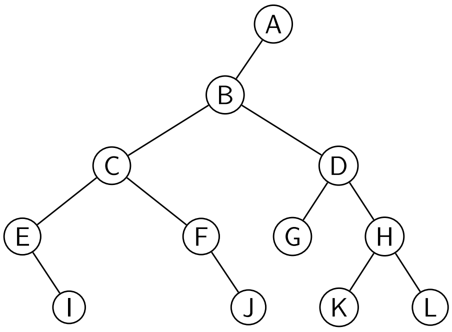
Suponha que **emOrdem** seja chamado em um nó x cuja subárvore esquerda tem k nós e cuja subárvore direita tem $n - k - 1$ nós. O tempo para executar essa função é limitado por $T(n) = T(k) + T(n - k - 1) + d$, para $d > 0$ refletindo o limite superior para o tempo de execução do corpo dessa função (print).

Exemplo



- Pré-ordem: ABDGCEHIF
- Em ordem: DGBAHEICF
- Pós-ordem: GDBHIEFCA
- largura: ABCDEFGHI

Exercício

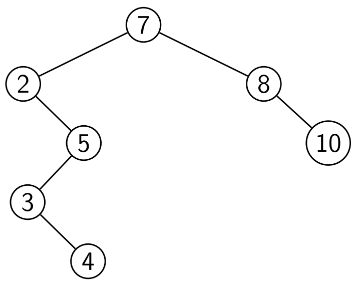


Aplicação

- importante toda vez que uma decisão binária deve ser tomada
- remoção de números repetidos em um conjunto não ordenado 7 8 2 5 8 3 5 10 4
  - compara cada novo número com um já lido
  - manter uma lista ordenada e a cada novo número percorrer a lista
  - manter os números em uma árvore binária

Aplicação

7 8 2 5 8 3 5 10 4

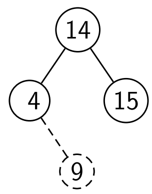


Construção para Classificar

- número é comparado com um nó
- se número for menor que o conteúdo do nó – inserí-lo em uma ramificação esquerda
- se número for maior ou igual ao conteúdo do nó – inserí-lo em uma ramificação direita
- 14 15 4 9 7 18 3 5 16 4 20 17 9 14 5

Construção para Classificar

14 15 4 9

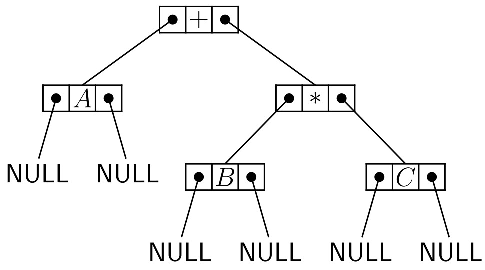


Após construir a árvore, vamos percorrer em pré-ordem.

Exercício

Percorra todas as árvores de expressão vistas nesta aula em pré-ordem, em ordem e pós-ordem.

Implementação



Bibliografia

- José Augusto Baranauskas, notas de aula da disciplina de *Algoritmos e Estruturas de Dados I*, Departamento de Física e Matemática-USP.
- Material de apoio para a disciplina de *Algoritmos e Estruturas de Dados I*, Departamento de Informática - Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa