

Árvores

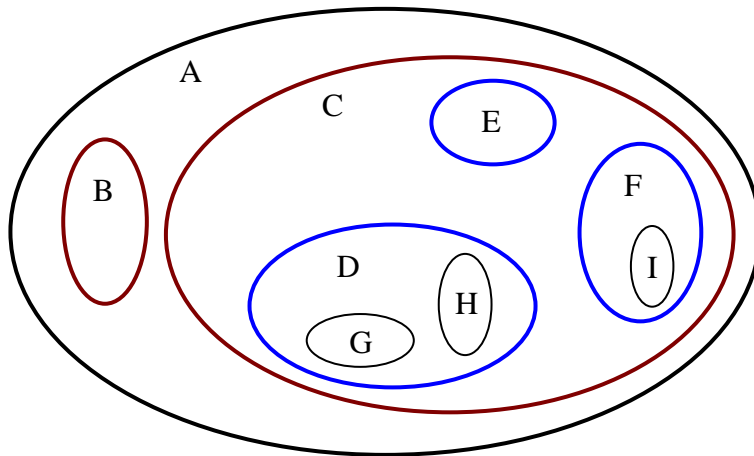
Representação gráfica

As três formas de representação gráfica são:

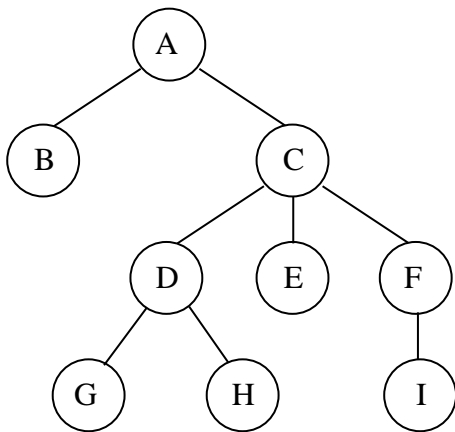
Representação por parênteses aninhados:

(A (B) (C (D (G) (H)) (E) (F (I))))

Diagrama de inclusão:



Representação hierárquica:



Motivação

- Diversas aplicações necessitam de estruturas mais complexas que as listas estudadas até agora.
- Inúmeros problemas podem ser modelados através de árvores.

Definição

Uma árvore T é um conjunto de elementos denominados nós ou vértices tais que:

- $T = \emptyset$, a árvore é dita vazia;
- existe um nó especial r , chamado **raiz de T** ; os restantes constituem um único conjunto vazio ou são divididos em m (deve ser maior ou igual a 1) conjuntos distintos não vazios que são as **subárvores de r** , cada subárvore a qual é, por sua vez, uma árvore.

Notação: T_v , se v é um nó de T então a notação T_v indica a **subárvore de T** com raiz em v .

Subárvore

Seja a árvore do exemplo anterior $T = \{A, B, \dots\}$

A árvore T possui duas subárvores:

T_b e **T_c** onde

$T_b = \{B\}$ e **$T_c = \{C, D, \dots\}$**

A subárvore **T_c** possui 3 subárvores:

T_d , **T_f** e **T_e** onde

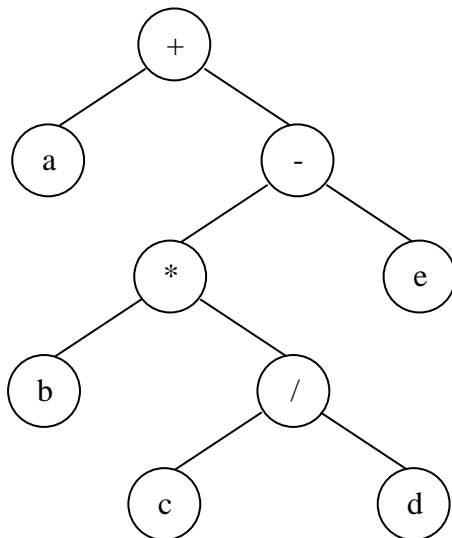
$T_d = \{D, G, H\}$

$T_f = \{F, I\}$

$T_e = \{E\}$

As subárvores **T_b** , **T_e** , **T_g** , **T_h** e **T_i** possuem apenas o nó raiz e nenhuma subárvore.

Exemplo da expressão aritmética: $(a + (b * (c / d) - e))$



Grau de saída, descendente e ancestral

O **número de filhos** de um nó é chamado **grau de saída** desse nó.

Se x pertence à subárvore T_v , então, x é descendente de v e v é **ancestral**, ou **antecessor**, de x . Se neste caso x é diferente de v então x é descendente próprio de v e v é ancestral próprio de x .

Nó folha e nó interior

Um nó que não possui descendentes próprios é chamado de nó folha, ou seja, nó folha é aquele com grau de saída nulo.

Um nó que não é folha (isto é, possui grau de saída diferente de zero) é chamado nó interior ou nó interno.

Nível (ou profundidade) e altura de um nó

O nível ou profundidade, de um nó é o número de nós do caminho da raiz até o nó.

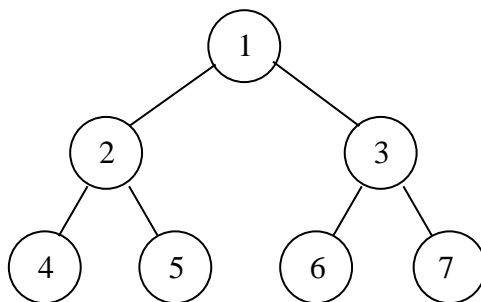
O nível da raiz é, portanto, 1.

A altura de um nó v é o número de nós no maior caminho de v até um de seus descendentes. As folhas têm altura 1.

Árvore Cheia

Uma árvore cheia possui o máximo de nós sucessores, isto é, todos os nós tem número máximo de filhos exceto as folhas, e todas as folhas estão na mesma altura.

Árvore cheia de grau 2: implementação sequencial.



Exemplo com Array (vetor)

Array com 7 posições:

1	2	3	4	5	6	7

Armazenamento por nível:

posição do nó	posição dos filhos do nó
1	2, 3
2	4, 5
3	6, 7
i	$(2i, 2i + 1)$

Árvores Binárias

Uma Árvore Binária T é um conjunto finito de elementos denominados nós ou vértices, tal que:

- $T = 0$, a árvore é dita vazia;
- Existe um nó especial r , chamado raiz de T , os restantes podem ser divididos em dois subconjuntos disjuntos, Tre e Trd , que são as subárvores esquerda e direita de r , respectivamente e as quais, por sua vez, também são árvores binárias.

Definição da Estrutura de Dados

```
Type tree = ^no
    no = record
        info: TpElemento;
        esq, dir: tree;
    end;
```

Operações associadas ao TAD árvore binária padrão:

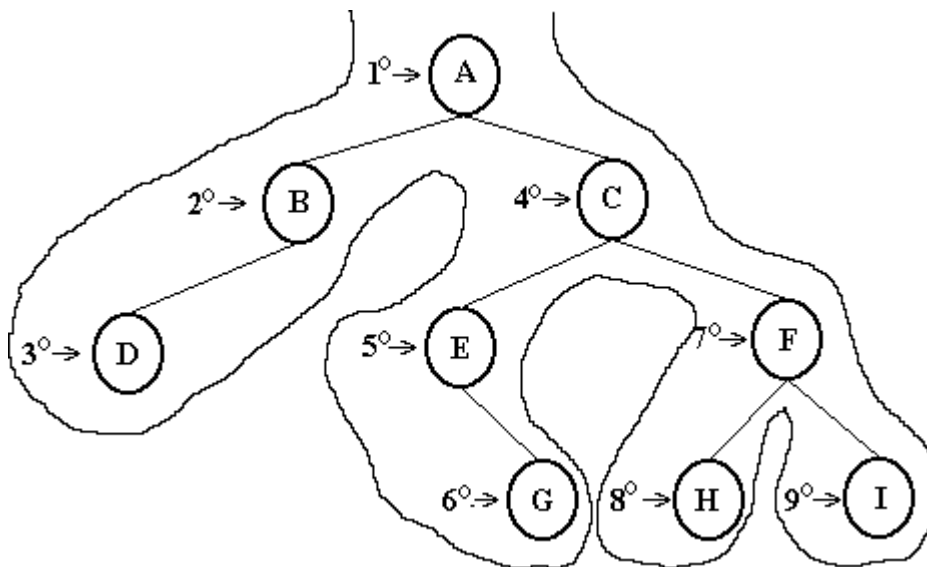
- Inicializar uma árvore binária;
- Criar um nó
- Verificar se a árvore está vazia ou não;
- Criar um filho à direita de um dado nó;
- Criar um filho à esquerda de um dado nó;
- Verificar qual o nível (ou profundidade) de um dado nó;
- Retornar o pai de um dado nó;

Percurso em Árvores Binárias

Há três maneiras recursivas de se percorrer árvores binárias:

Travessia em Pré-Ordem

1. se árvore vazia; fim
2. exibir o campo info do nó
3. percorrer em pré-ordem a subárvore esquerda
4. percorrer em pré-ordem a subárvore direita



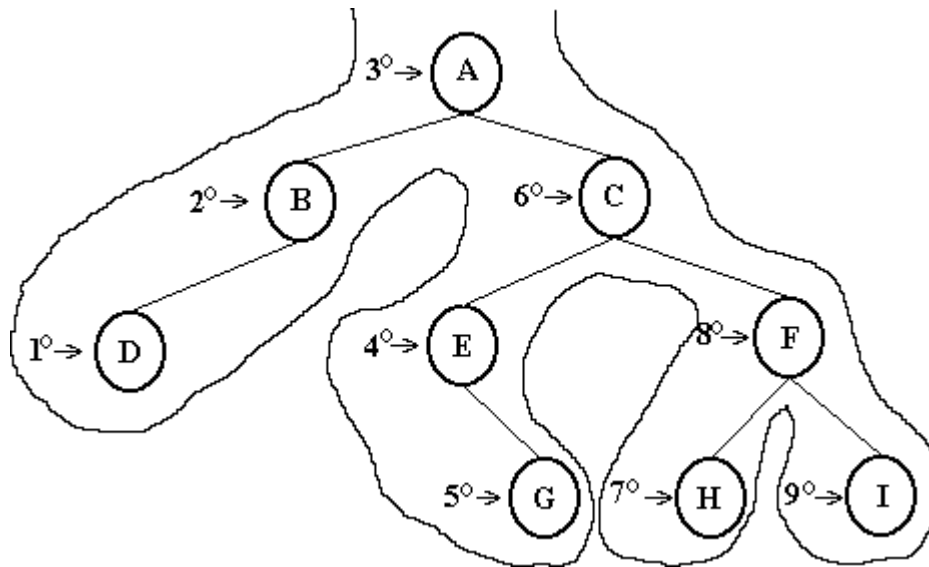
Resultado do Pré-Ordem:

ABDCEGFHI

visita o nó (exibe o info) quando passar a sua esquerda

Travessia em In-Ordem

1. se árvore vazia; fim
2. percorrer em in-ordem a subárvore esquerda
3. exibir o campo info do nó
4. percorrer em in-ordem a subárvore direita



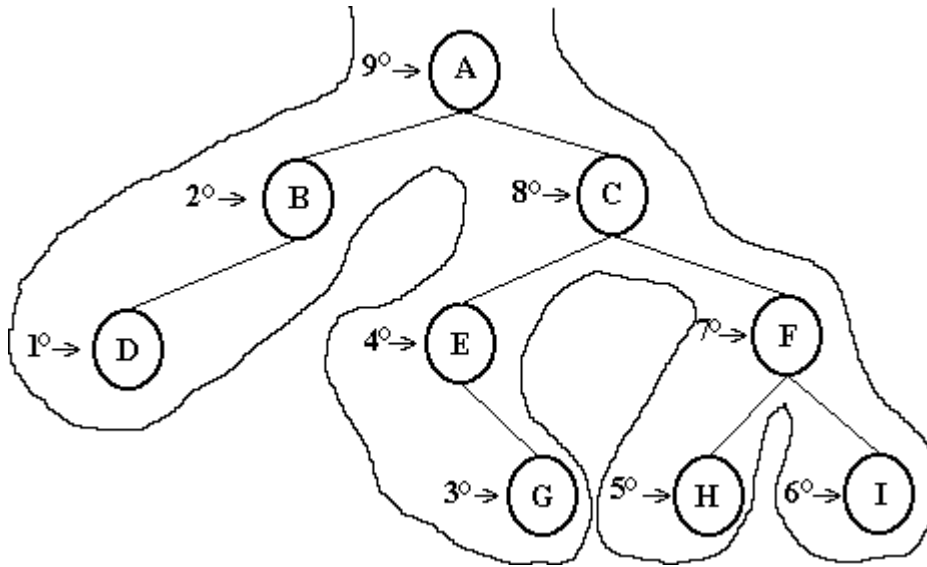
Resultado do In-Ordem:

DBAEGCHFI

visita o nó (exibe o info) quando passar embaixo do nó

Travessia em Pós-Ordem

1. se árvore vazia, fim
2. percorrer em Pós-Ordem a subárvore esquerda
3. percorrer em Pós-Ordem a subárvore direita
4. exibir o campo info do nó



Resultado do Pós-Ordem:

DBGEHIFCA

visita o nó (exibe o info) quando passar a sua direita

Algoritmos a serem implementados:

- 1) Utilizar os algoritmos de travessia para fazer uma busca na Árvore (localizar um determinado **elemento** passado por parâmetro, retornando: achou **true (1)** ou **false (0)**, e se achou, retornar o nodo que se encontra o elemento).
- 2) Tornar uma árvore Vazia (**free** em todos os nodos da árvore).
- 3) Verificar qual o nível (ou profundidade) de um dado nó.
- 4) Retornar o pai de um dado nó.
- 5) Exibe a árvore no formato de árvore.