

ALGORITMOS E ESTRUTURAS DE DADOS 2023/2024

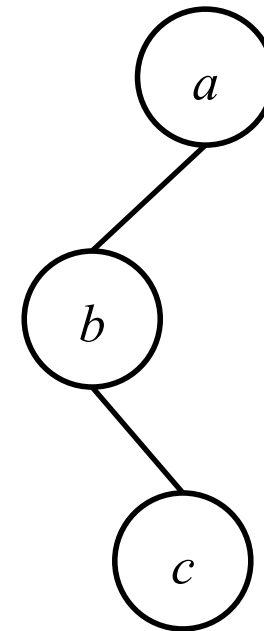
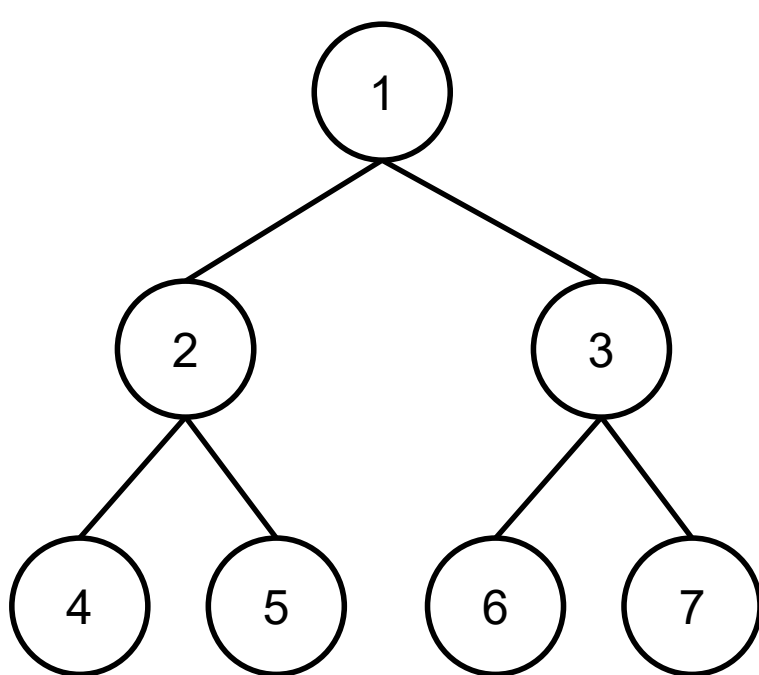
ÁRVORE BINÁRIA

Armanda Rodrigues

27 de outubro de 2023

Relação entre a altura e o número de nós

- O número de nós de uma árvore binária de altura h varia entre h e $2^h - 1$
- A altura de uma árvore binária com n nós varia entre $\lceil \log(n+1) \rceil$ e n



Classe Nó de Árvore Binária (1)

```
package dataStructures;

import java.io.Serializable;

class BTNode<E> implements Serializable{

    // Element stored in the node.
    private E element;

    // (Pointer to) the left child.
    private BTNode<E> leftChild;

    // (Pointer to) the right child.
    private BTNode<E> rightChild;

    ...
}
```

Classe Nó de Árvore Binária (2)

```
public BTreeNode( E theElement, BTreeNode<E> theLeft, BTreeNode<E> theRight ){  
    element = theElement;  
    leftChild = theLeft;  
    rightChild = theRight;  
}  
  
public BTreeNode( E theElement ){  
    this(theElement, null, null);  
}
```

Classe Nó de Árvore Binária (3)

```
public E getElement( ){  
    return element;  
}
```

```
public BTNode<E> getLeft( ){  
    return leftChild;  
}
```

```
public BTNode<E> getRight( ){  
    return rightChild;  
}
```

Classe Nó de Árvore Binária (4)

```
public void setElement( E newElement ){
    element = newElement;
}

public void setLeft( BTNode<E> newLeft ){
    leftChild = newLeft;
}

public void setRight( BTNode<E> newRight ){
    rightChild = newRight;
}
```

Classe Nó de Árvore Binária (5)

```
// Returns true iff the node is a leaf.  
public boolean isLeaf( ){  
    return leftChild == null && rightChild == null;  
}  
} // End of BTreeNode.
```

se for uma folha não tem nenhum filho

Árvore Binária

```
// The root of the tree.  
protected BTreeNode<E> root;
```

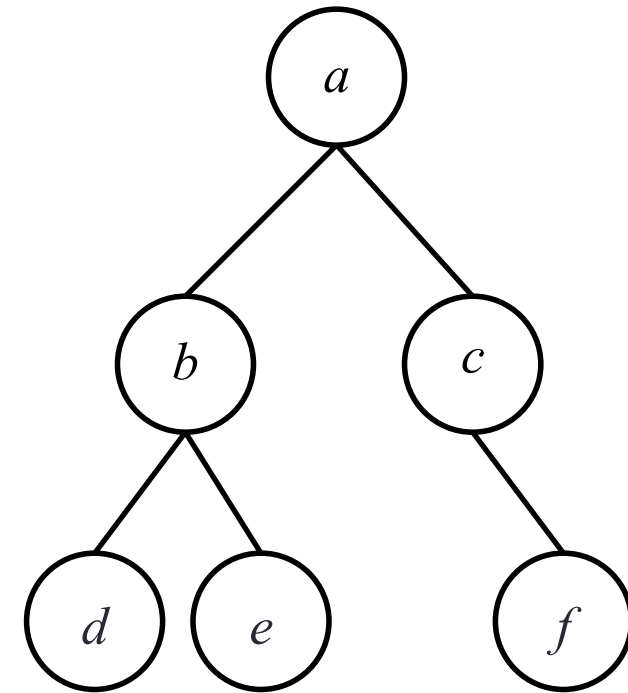
```
.....
```

```
// Returns true iff the binary tree contains no elements.  
public boolean isEmpty( ){  
    return root == null;  
}
```

```
.....
```


Percursos de Árvore Binária

- Percorrer toda a árvore, visitando todos os nós
- Utilizar recursividade
- 3 percursos diferentes, a ordem de visita dos nós é alterada
 - Percurso Prefixo
 - Percurso Infixo
 - Percurso Sufixo

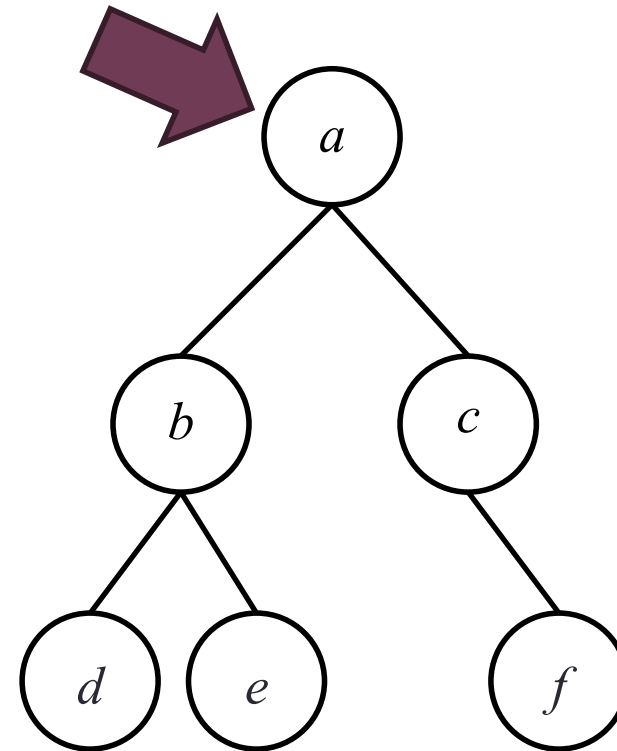


Esquema do Percurso Prefixo (recursivo)

```
public void preorder( ){  
    this.preorder(root);  
}  
  
protected void preorder( BTNode<E> node ){  
    if ( node != null ){  
        PROCESS( node.getElement() );  
        this.preorder( node.getLeft() );  
        this.preorder( node.getRight() );  
    }  
}
```

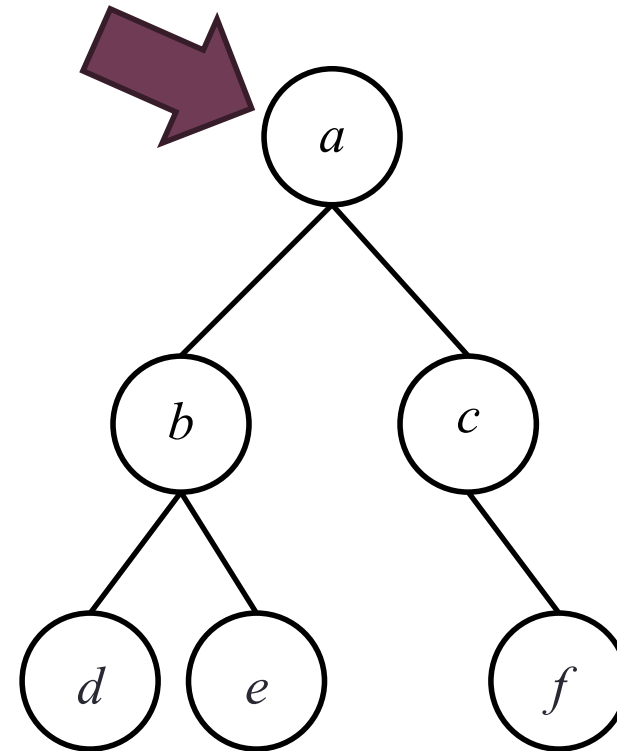
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
3. Desce para a sub-árvore direita



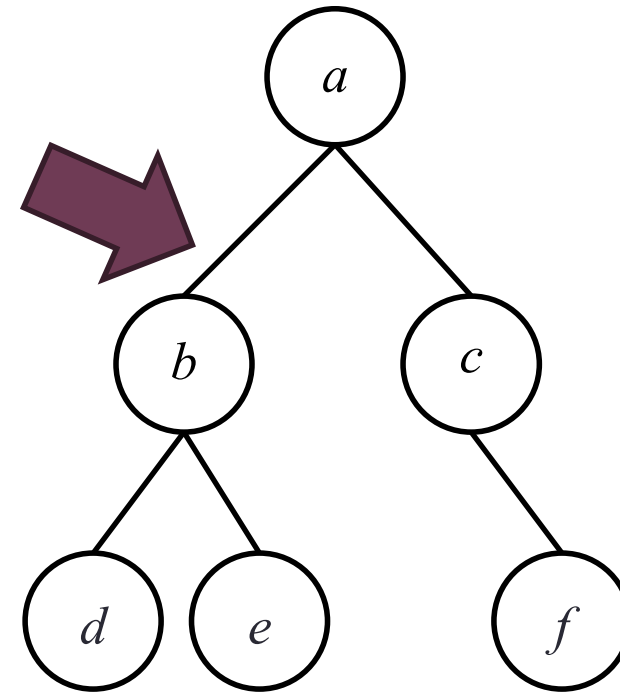
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
3. Desce para a sub-árvore direita



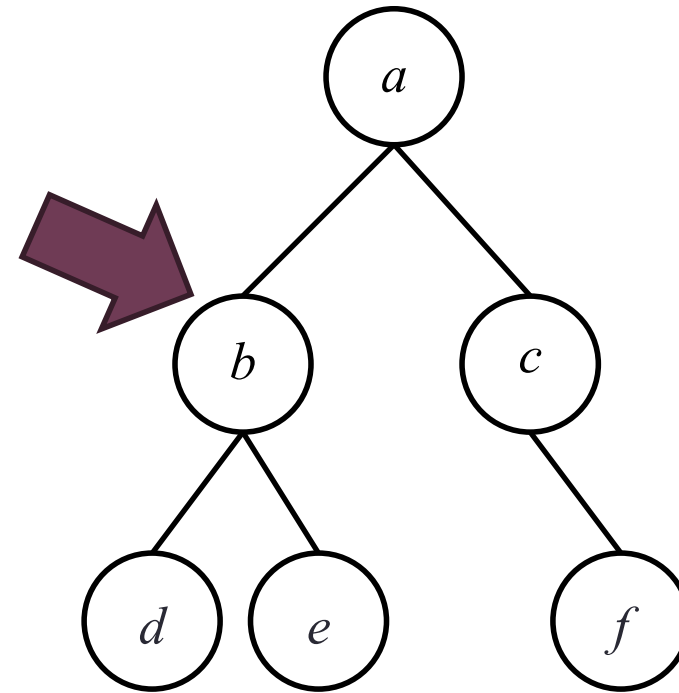
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
3. Desce para a sub-árvore direita



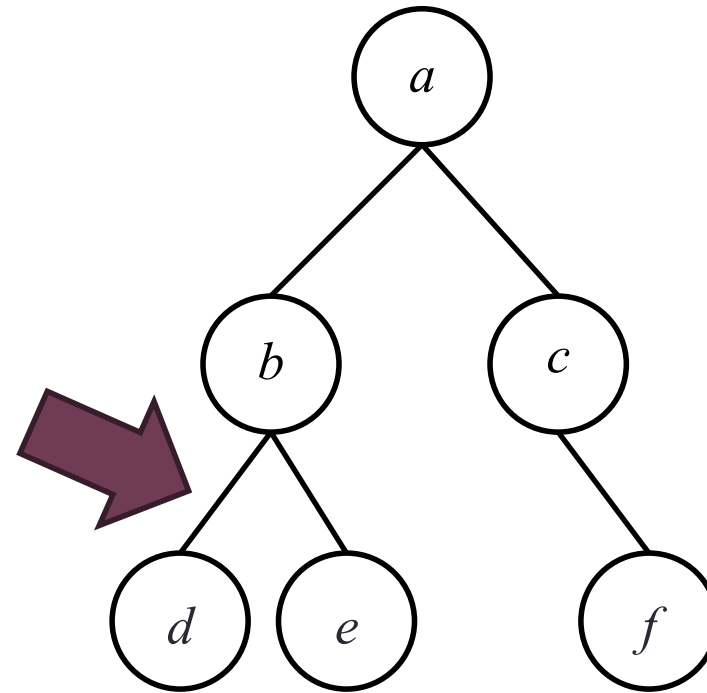
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b***
 - 2.2 Desce para a sub-árvore esquerda
 - 2.3 Desce para a sub-árvore direita
3. Desce para a sub-árvore direita



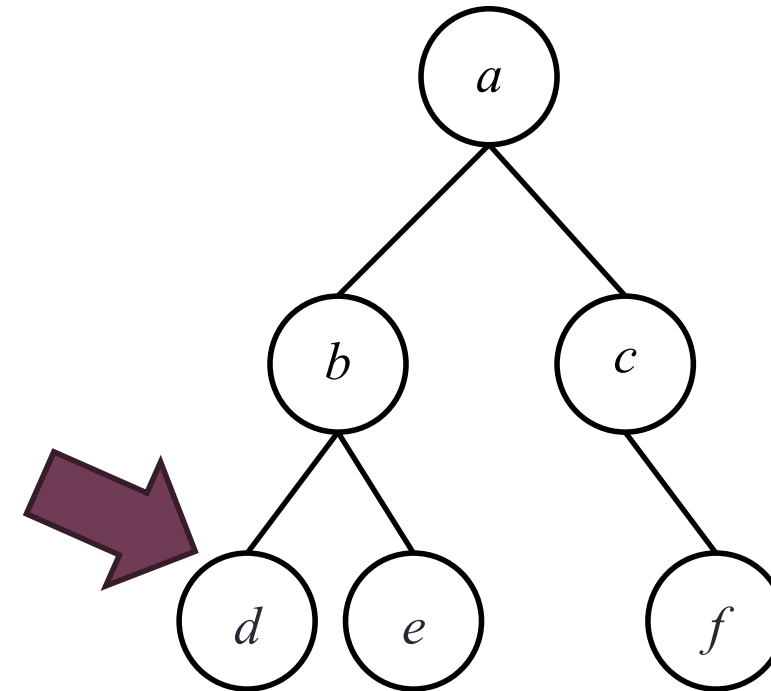
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda**
 - 2.3 Desce para a sub-árvore direita
3. Desce para a sub-árvore direita



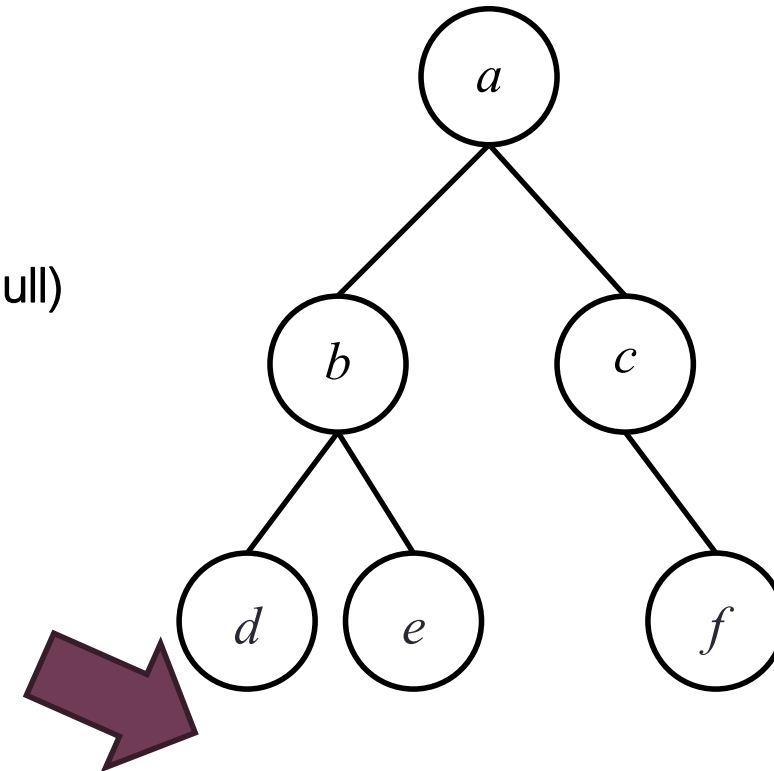
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d***
 - 2.2.2 Desce para a sub-árvore Esquerda
 - 2.2.3 Desce para a sub-árvore Direita
 - 2.3 Desce para a sub-árvore direita
3. Desce para a sub-árvore direita



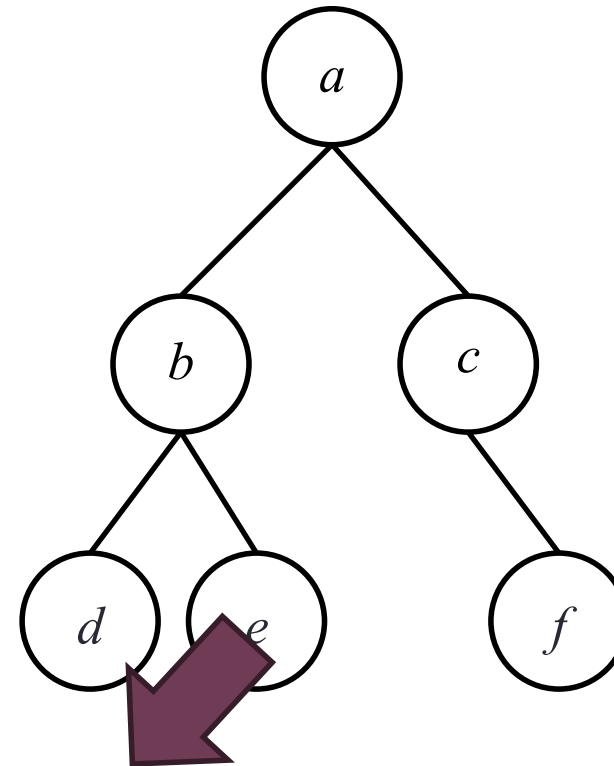
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)**
 - 2.2.3 Desce para a sub-árvore Direita
 - 2.3 Desce para a sub-árvore direita
3. Desce para a sub-árvore direita



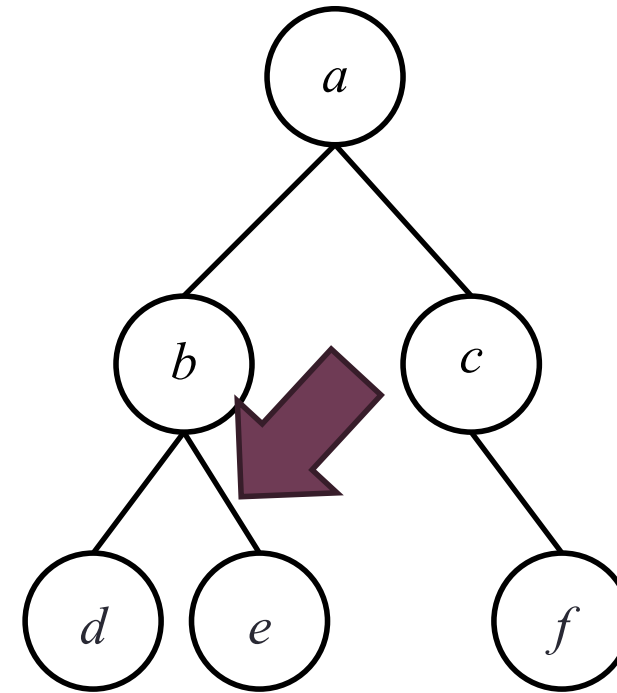
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)**
 - 2.3 Desce para a sub-árvore direita
3. Desce para a sub-árvore direita



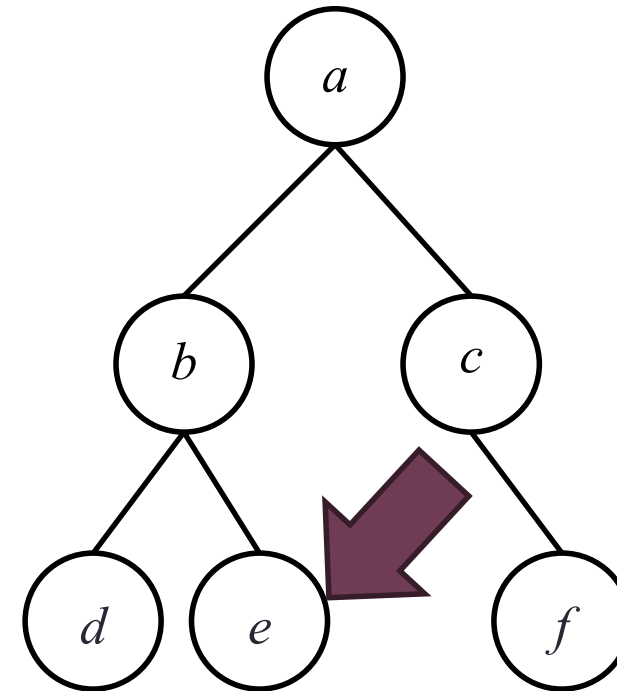
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita**
3. Desce para a sub-árvore direita



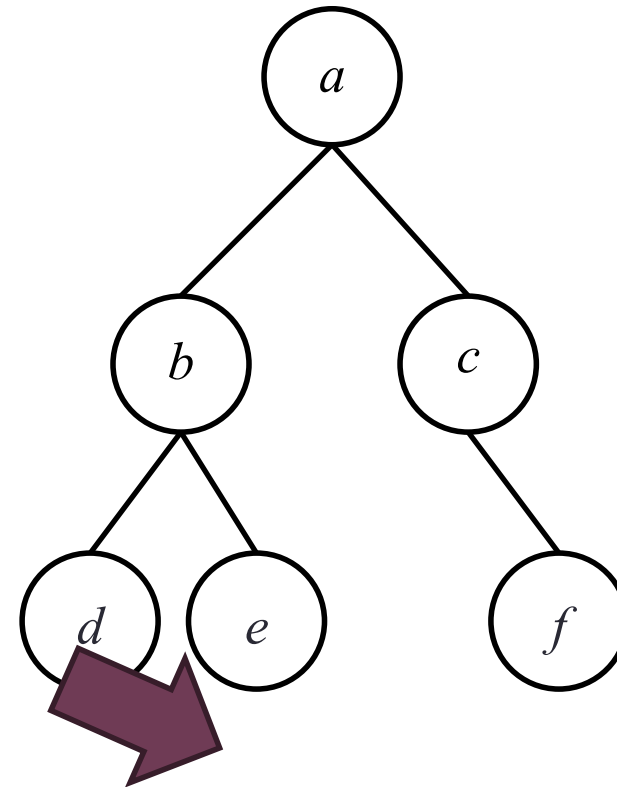
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d*
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita *e***
 - 2.3.2 Desce para a sub-árvore Esquerda
 - 2.3.3 Desce para a sub-árvore Direita
3. Desce para a sub-árvore direita



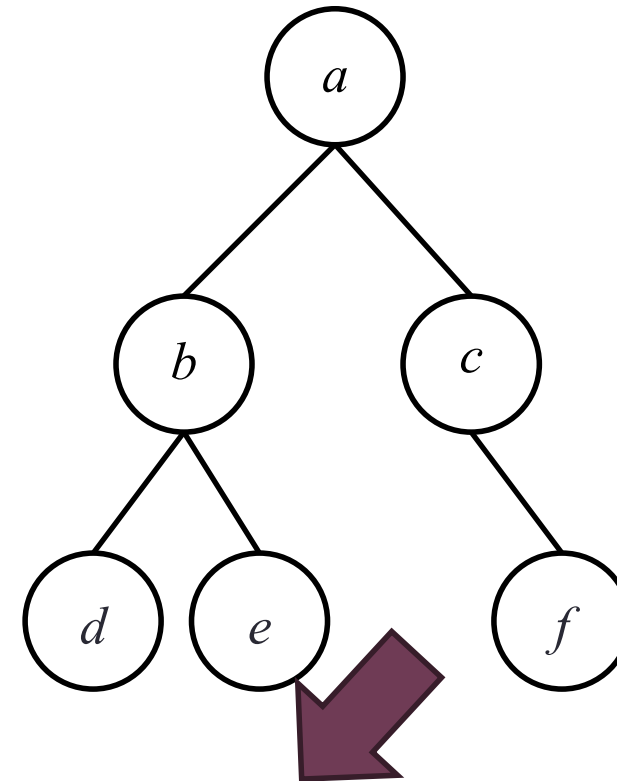
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita e
 - 2.3.2 Desce para a sub-árvore Esquerda (null)**
 - 2.3.3 Desce para a sub-árvore Direita
3. Desce para a sub-árvore direita



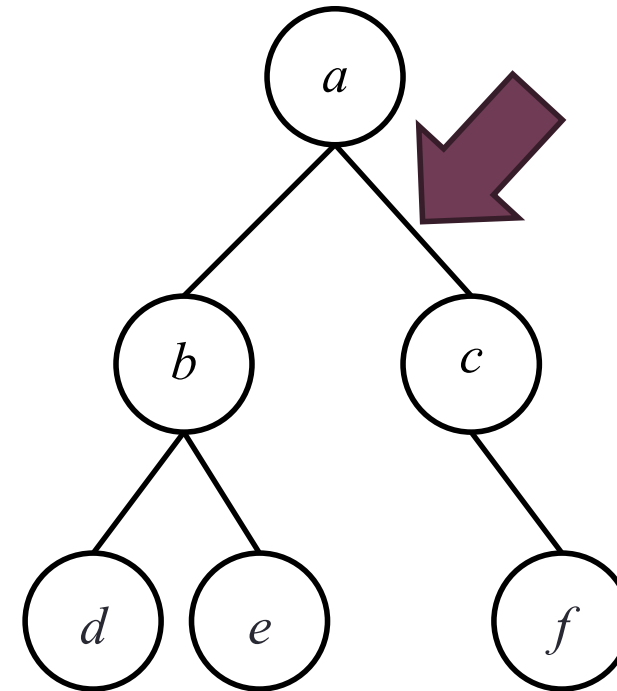
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita e
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)**
3. Desce para a sub-árvore direita



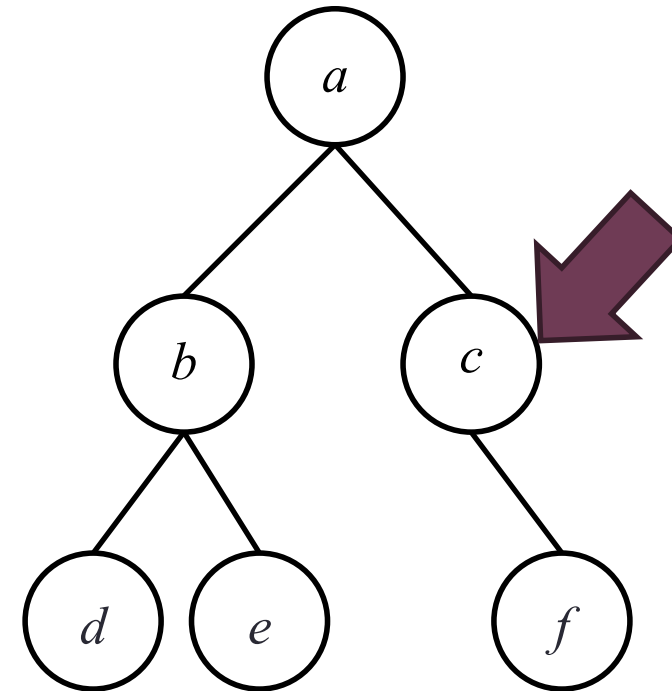
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita e
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita



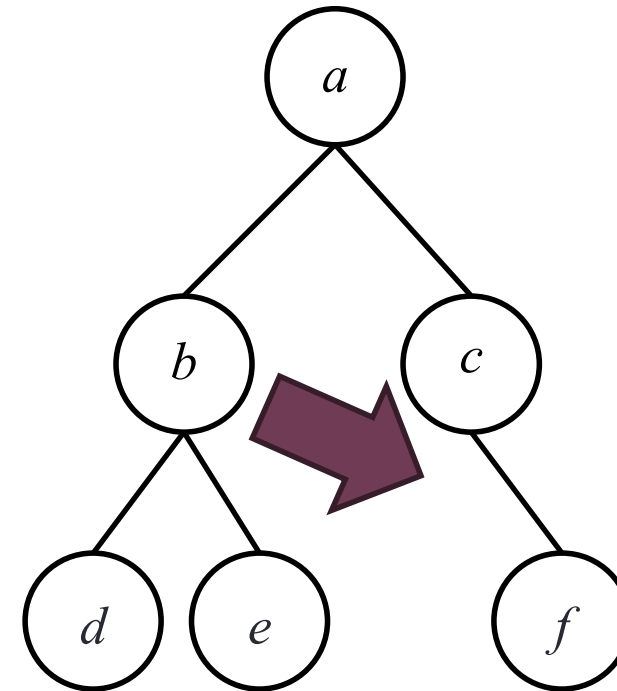
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d*
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita *e*
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 **Visita *c***
 - 3.2 Desce para a sub-árvore Esquerda
 - 3.3 Desce para a sub-árvore Direita



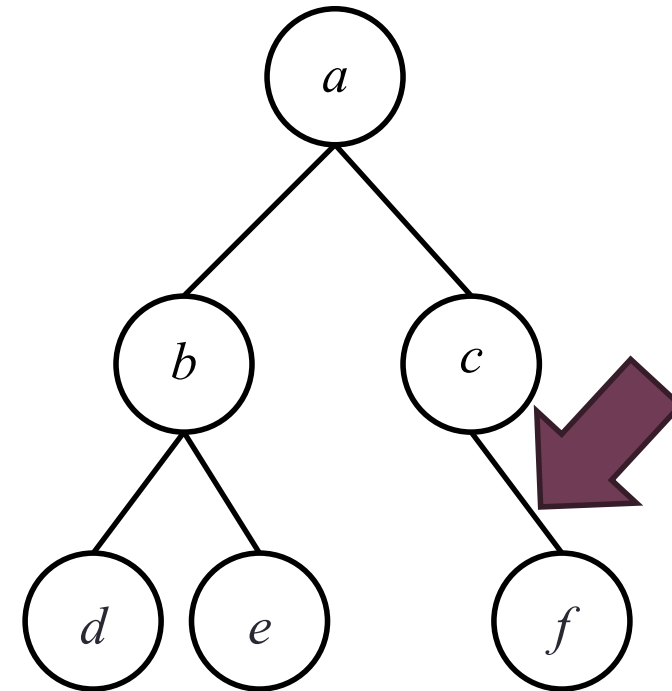
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d*
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita *e*
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 Visita *c*
 - 3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3 Desce para a sub-árvore Direita



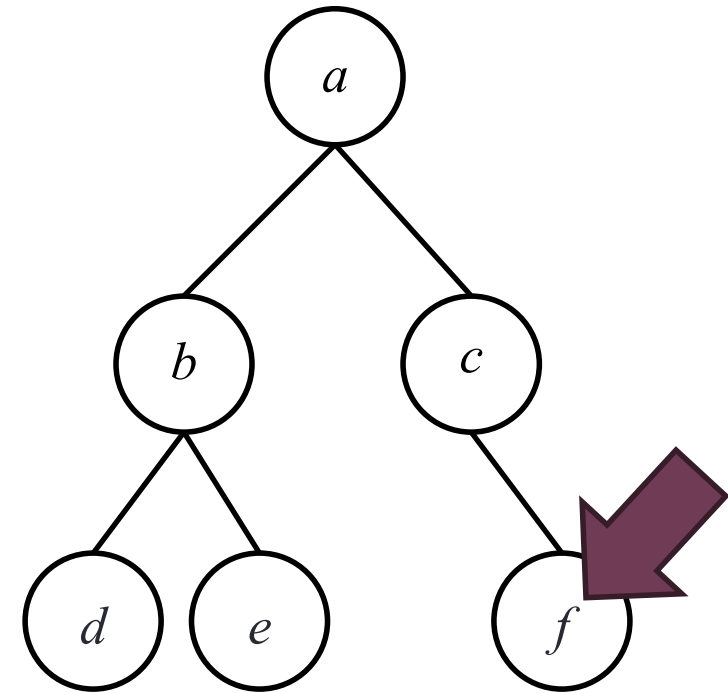
Percurso Prefixo

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita e
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 Visita c
 - 3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3 Desce para a sub-árvore Direita



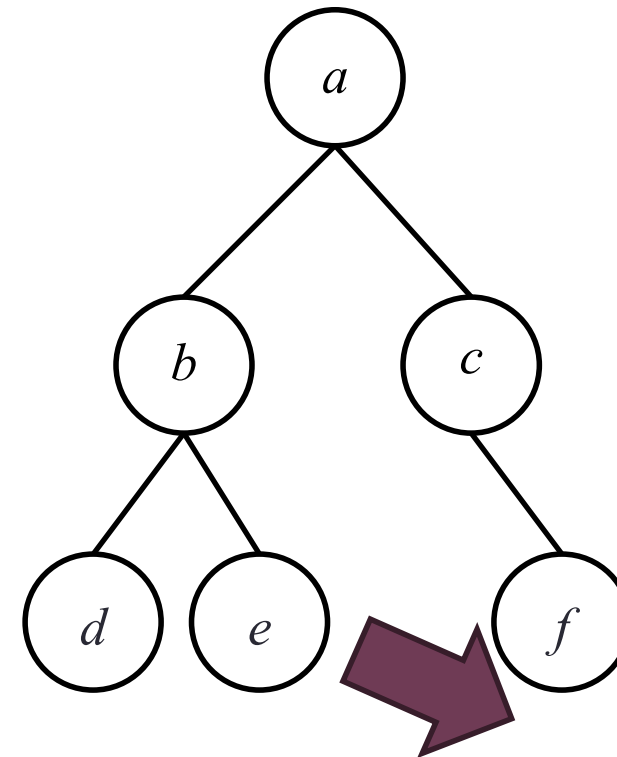
Percurso Prefixo

1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d*
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita *e*
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 Visita *c*
 - 3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3 Desce para a sub-árvore Direita
 - 3.3.1 Visita *f***
 - 3.3.2 Desce para a sub-árvore Esquerda
 - 3.3.3 Desce para a sub-árvore Direita



Percurso Prefixo

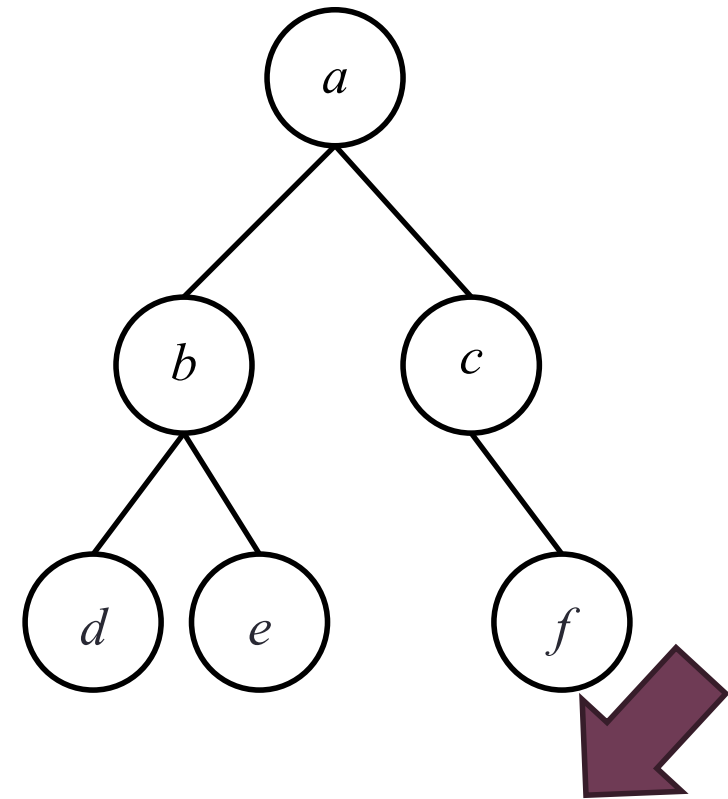
1. Visita *a*
2. Desce para a sub-árvore esquerda
 - 2.1 Visita *b*
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita *d*
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita *e*
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 Visita *c*
 - 3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3 Desce para a sub-árvore Direita
 - 3.3.1 Visita *f*
 - 3.3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3.3 Desce para a sub-árvore Direita



Percurso Prefixo

Percurso Prefixo: a, b, d, e, c, f

1. Visita a
2. Desce para a sub-árvore esquerda
 - 2.1 Visita b
 - 2.2 Desce para a sub-árvore esquerda
 - 2.2.1 Visita d
 - 2.2.2 Desce para a sub-árvore Esquerda (null)
 - 2.2.3 Desce para a sub-árvore Direita (null)
 - 2.3 Desce para a sub-árvore direita
 - 2.3.1 Visita e
 - 2.3.2 Desce para a sub-árvore Esquerda (null)
 - 2.3.3 Desce para a sub-árvore Direita (null)
3. Desce para a sub-árvore direita
 - 3.1 Visita c
 - 3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3 Desce para a sub-árvore Direita
 - 3.3.1 Visita f
 - 3.3.2 Desce para a sub-árvore Esquerda (null)
 - 3.3.3 Desce para a sub-árvore Direita (null)



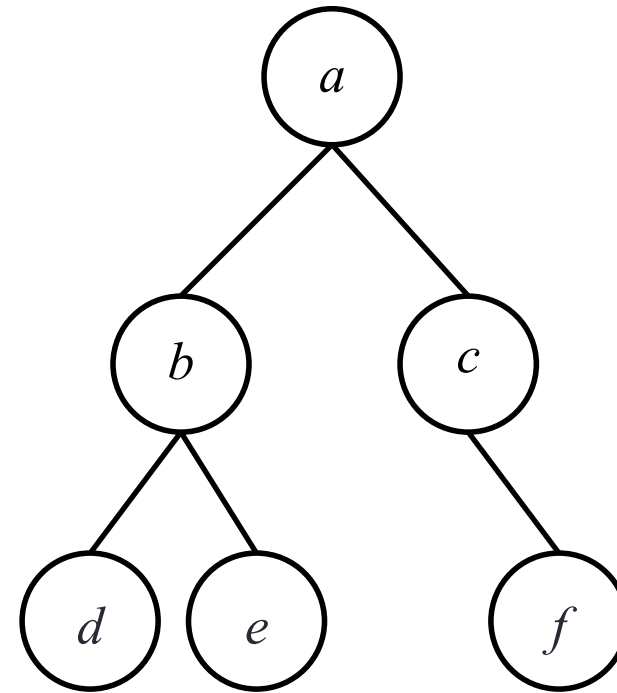
Esquema do Percurso Infixo (recursivo)

```
public void inorder( ){  
    this.inorder(root);  
}  
  
protected void inorder( BTNode<E> node ){  
    if ( node != null ){  
        this.inorder( node.getLeft() );  
        PROCESS( node.getElement() );  
        this.inorder( node.getRight() );  
    }  
}
```

???

Percorso Infixo

Percorso Infixo: d, b, e, a, c, f

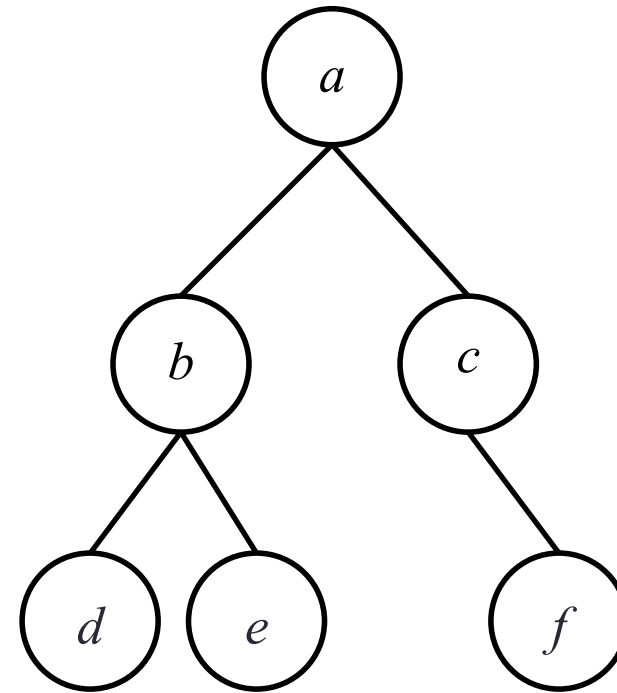


Esquema do Percurso Sufixo (recursivo)

```
public void postorder( ){  
    this.postorder(root);  
}  
  
protected void postorder( BTreeNode<E> node ){  
    if ( node != null ){  
        this.postorder( node.getLeft() );  
        this.postorder( node.getRight() );  
        PROCESS( node.getElement() );  
    }  
}
```


Percurso Sufixo

Percurso Sufixo: d, e, b, f, c, a



Complexidades dos Percursos Recursivos

- Sendo n o número de nós da árvore, a complexidade do percurso prefixo, infixo ou sufixo

$O(n) \times \text{"custo de PROCESS(elemento)"}$,

no melhor caso, no pior caso e no caso esperado.

- Justificação
 - O número de chamadas a *preorder*, *inorder* ou *postorder* é igual ao número total de sub-árvores.
 - A árvore tem n sub-árvores não vazias.
 - A árvore tem $n+1$ sub-árvores vazias.
 - Logo, a árvore tem $2n+1 = O(n)$ sub-árvores.