



Entendendo os Percursos em Árvores

Serão apresentados a seguir os percursos em árvores binárias: em ordem, em pós ordem e em pré ordem. E a implementação desses percursos na linguagem de programação.

EM ORDEM

No percurso em ordem, o percurso pela árvore binária é realizado na ordem, ou seja, esquerda, raiz e direita.

emOrdem (BIntNo ABB)

início

se (ABB <> nulo)

então

emOrdem(ABB.esquerda);

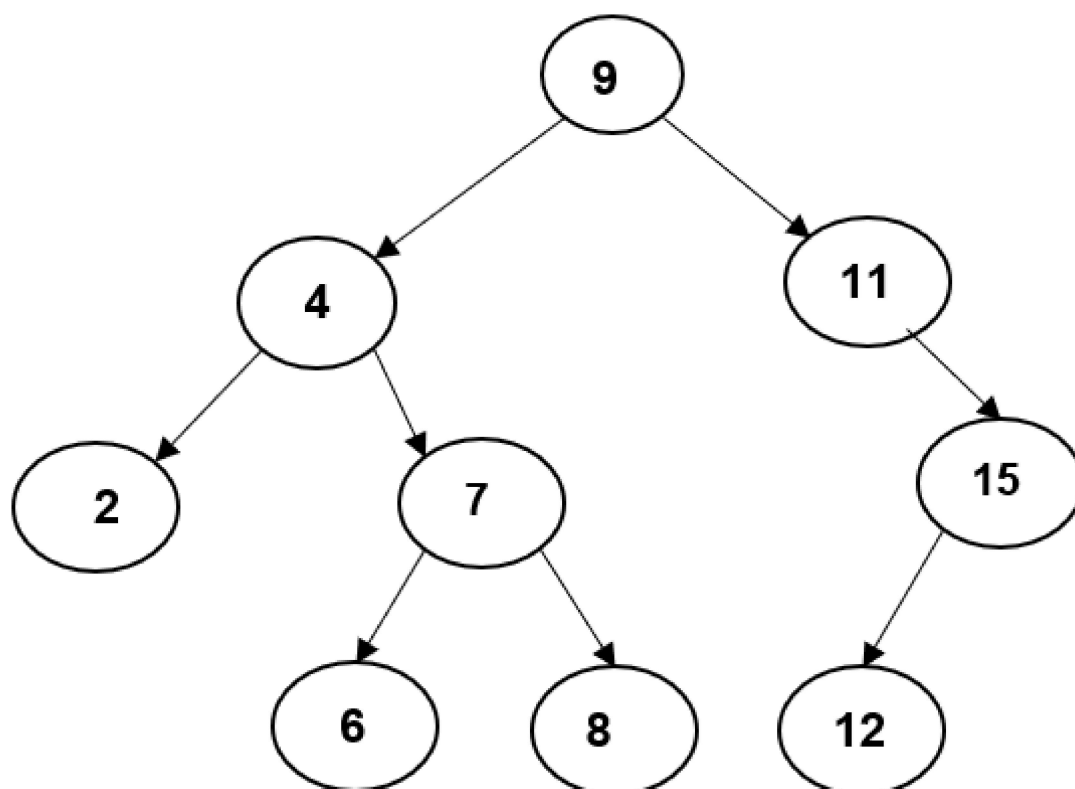
visita(ABB);

emOrdem(ABB.direita);

fimse;

fim.

Observe a figura que representa uma árvore binária:



O percurso em ordem desta árvore binária será:

2 4 6 7 8 9 11 12 15

PÓS ORDEM

No percurso pós ordem, o percurso pela árvore binária é realizado em pós ordem, ou seja, esquerda, direita e raiz.

posOrdem (BIntNo ABB)

início

se (ABB <> nulo)

então

posOrdem(ABB.esquerda);



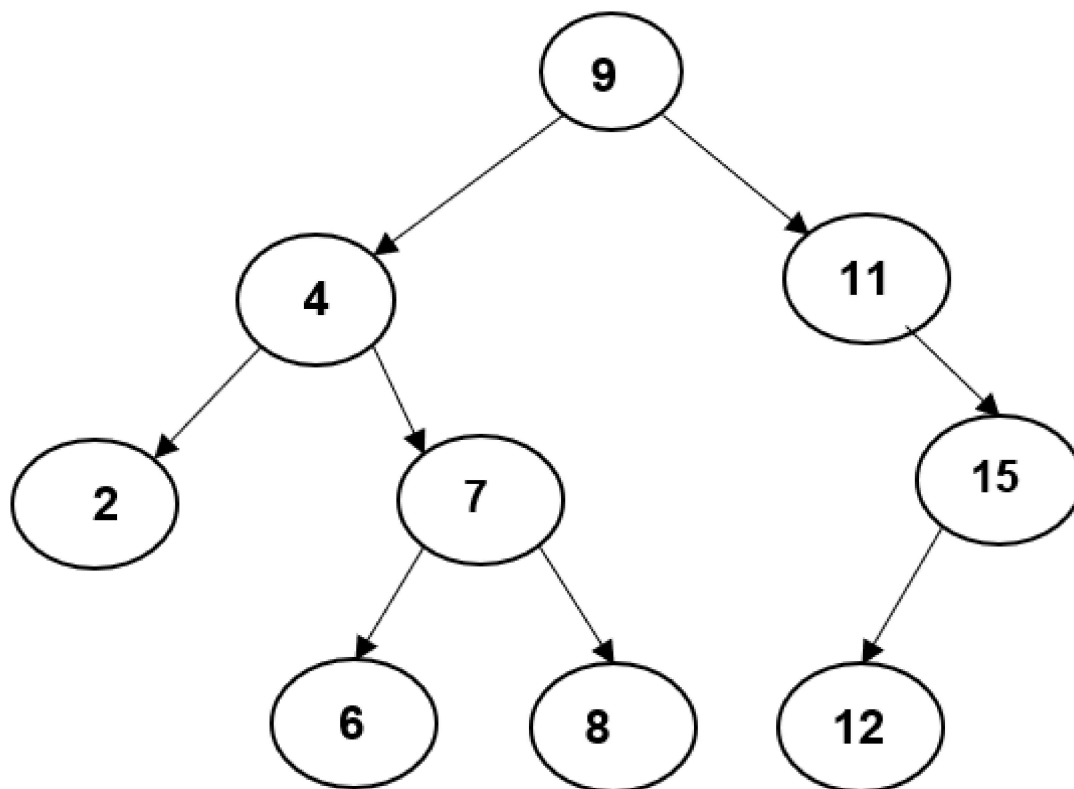
posOrdem(ABB.direita);

visita(ABB);

fimse;

fim.

Observe a figura que representa uma árvore binária.



O percurso pós ordem desta árvore binária será:

2 6 8 7 4 12 15 11 9

PRÉ ORDEM

No percurso pré-ordem, o percurso pela árvore binária é realizado em pré-ordem, ou seja, raiz, esquerda e direita.



preOrdem (BIntNo ABB)

início

se (ABB <> nulo)

então

visita(ABB);

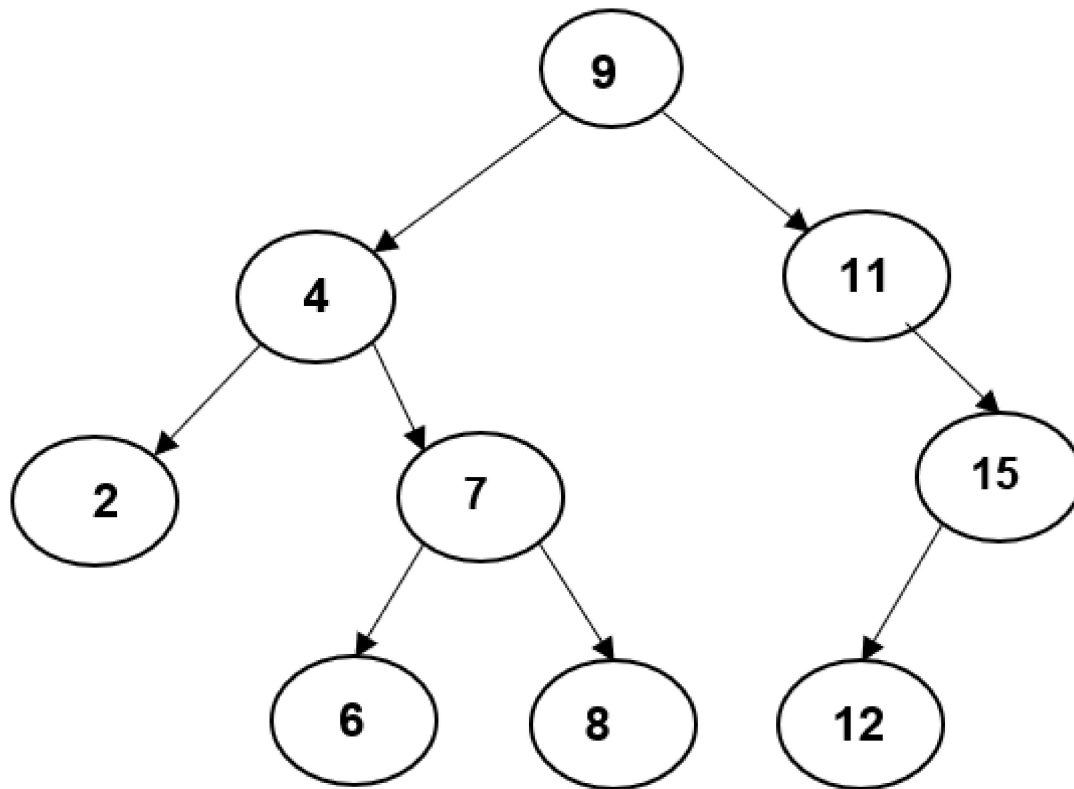
preOrdem(ABB.esquerda);

preOrdem(ABB.direita);

fimse;

fim.

Observe a figura que representa uma árvore binária.



O percurso pré ordem da árvore binária será:

9 4 2 7 6 8 11 15 12

IMPLEMENTANDO PERCURSOS NO JAVA

No percurso em ordem, o percurso pela árvore binária é realizado na ordem, ou seja, esquerda, raiz e direita. Na linguagem Java, o método recursivo do percurso em ordem pode ser escrito como segue.

```
emOrdem (BIntNo ABB)
```

```
{
```

```
    if (ABB != null)
```

```
{
```

```
emOrdem(ABB.esquerda);
```



```
visita(ABB); // é um método que mostra as informações do nó
```

```
emOrdem(ABB.direita);
```

```
}
```

```
}
```

No percurso pós ordem, o percurso pela árvore binária é realizado em pós ordem, ou seja, esquerda, direita e raiz. Na linguagem Java, o método recursivo do percurso pós ordem pode ser escrito como segue.

```
posOrdem (BIntNo ABB)
```

```
{
```

```
if (ABB != null)
```

```
{
```

```
posOrdem(ABB.esquerda);
```

```
posOrdem(ABB.direita);
```

```
visita(ABB); // é um método que mostra as informações do nó
```

```
}
```

```
}
```

No percurso pré-ordem, o percurso pela árvore binária é realizado em pré-ordem, ou seja, raiz, esquerda e direita. Na linguagem Java, o método recursivo do percurso pré ordem pode ser escrito como segue.

preOrdem (BIntNo ABB)



```
{  
  
    if (ABB != null)  
  
        então  
  
            visita(ABB); // é um método que mostra as informações do nó  
  
            preOrdem(ABB.esquerda);  
  
            preOrdem(ABB.direita);  
  
}  
  
}
```

Atividade extra

Indicação de leitura:

Você pode utilizar o livro Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++, da Ana Fernanda Gomes Ascencio e Graziela Santos de Araújo, no capítulo 7 sobre as aplicações das Árvores Binárias.

Referência Bibliográfica

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos Teoria e Prática**. Editora Cammpus. 3a Edição. 2012.



Atividade Prática 15 – Entendendo os percursos em Árvores

Título da Prática: Aplicações com os percursos em Árvores em Java

Objetivos: Entender como utilizar o netbeans para desenvolver programas em Java para manipular e aplicar os percursos em árvores

Materiais, Métodos e Ferramentas: Computador, netbeans, Java.

Atividade Prática

Os algoritmos de percursos em árvores podem ser escritos como segue.

Desenvolva o programa em Java deste algoritmo no NetBeans.

. emOrdem (BlntNo ABB)

início

se (ABB <> nulo)

então

emOrdem(ABB.esquerda);

visita(ABB);



emOrdem(ABB.direita);

fimse;

fim.

posOrdem (BIntNo ABB)

início

se (ABB <> nulo)

então

posOrdem(ABB.esquerda);

posOrdem(ABB.direita);

visita(ABB);

fimse;

fim.

preOrdem (BIntNo ABB)

início

se (ABB <> nulo)

então

visita(ABB);



preOrdem(ABB.esquerda);

preOrdem(ABB.direita);

fimse;

fim.

—

Gabarito Atividade Prática

emOrdem (BIntNo ABB)

{

if (ABB != null)

{

emOrdem(ABB.esquerda);

visita(ABB); // é um método que mostra as informações do nó

emOrdem(ABB.direita);



```
}
```

```
}
```

```
posOrdem (BIntNo ABB)
```

```
{
```

```
if (ABB != null)
```

```
{
```

```
    posOrdem(ABB.esquerda);
```

```
    posOrdem(ABB.direita);
```

```
    visita(ABB); // é um método que mostra as informações do nó
```

```
}
```

```
}
```

```
preOrdem (BIntNo ABB)
```

```
{
```

```
if (ABB != null)
```

```
    então
```

```
        visita(ABB); // é um método que mostra as informações do nó
```

```
preOrdem(ABB.esquerda);
```



```
preOrdem(ABB.direita);
```

```
}
```

```
}
```

[Ir para exercício](#)