unipac.br
Barbacena

Bacharelado em Ciência da Computação

Estruturas de Dados

Material de Apoio

Parte V – *Árvores*

Prof. Nairon Neri Silva
naironsilva@unipac.br

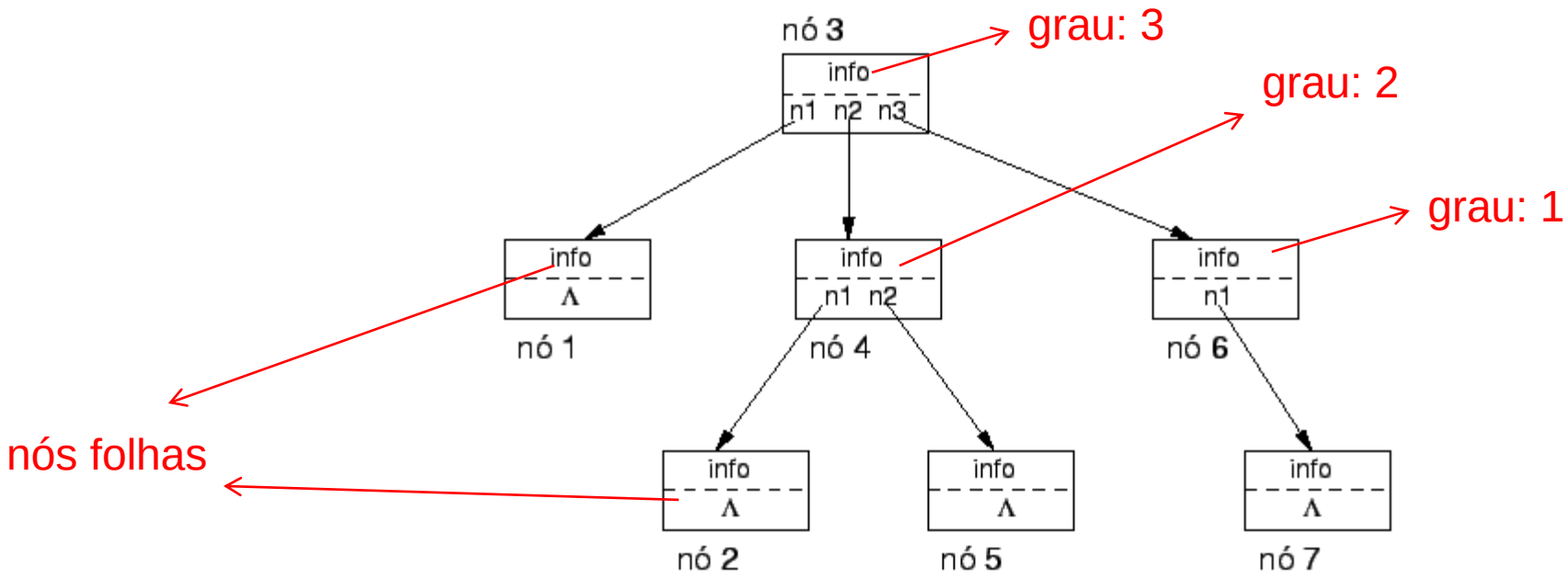
2º sem / 2021

Árvores

- Árvore é uma estrutura extensivamente utilizada na programação de sistemas, que esquematicamente pode ser visualizada como uma extensão de uma lista encadeada na qual um nó pode ter mais de um sucessor.
- É uma estrutura que contém um conjunto finito de um ou mais nós, sendo que um dos nós é especialmente designado como o **nó raiz,** e os demais nós são particionados em 0 ou mais conjuntos disjuntos onde cada um desses conjuntos é em si uma árvore, que recebe o nome de sub-árvore.

Árvores

- A representação esquemática de árvores usualmente coloca a raiz no topo, com a árvore crescendo para baixo.
- Grau do nó: número de sub-árvores de um nó.

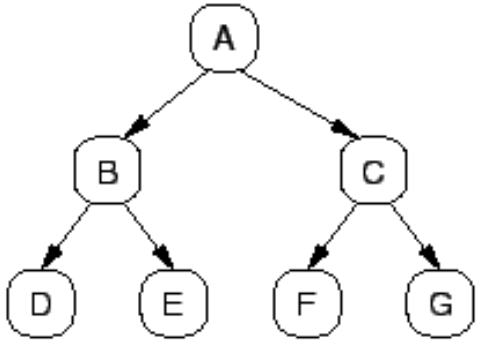


Varredura

- Os nós de uma árvore podem ser visitados em muitas ordens diferentes. Cada ordem define uma varredura da árvore.
- Na varredura e-r-d (esquerda-raiz-direita), visitamos:
  - a subárvore esquerda da raiz, em ordem e-r-d
  - a raiz
  - a subárvore direita da raiz, em ordem e-r-d

Árvore binária

- Um tipo especial de árvore é a ***árvore binária***. Uma árvore binária tem um nó raiz e no máximo duas sub-árvores (uma sub-árvore esquerda e uma sub-árvore direita).
- Em decorrência dessa definição, o grau de uma árvore binária é limitado a dois.



Árvore binária

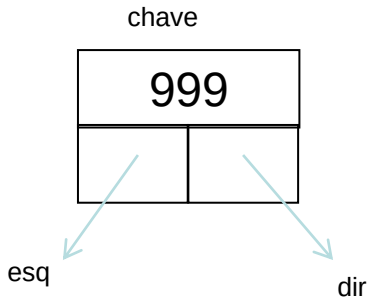
- Árvore binária é uma estrutura de dados mais geral que uma lista encadeada.
- É um conjunto A de nós (registros/células), tal que:
  - os filhos de cada elemento de A pertencem a A
  - todo elemento de A tem no máximo um pai
  - um e apenas um elemento de A não tem pai (raiz)
  - os filhos esquerdo e direito de cada elemento são distintos
  - não há ciclos

Árvore binária - Implementação

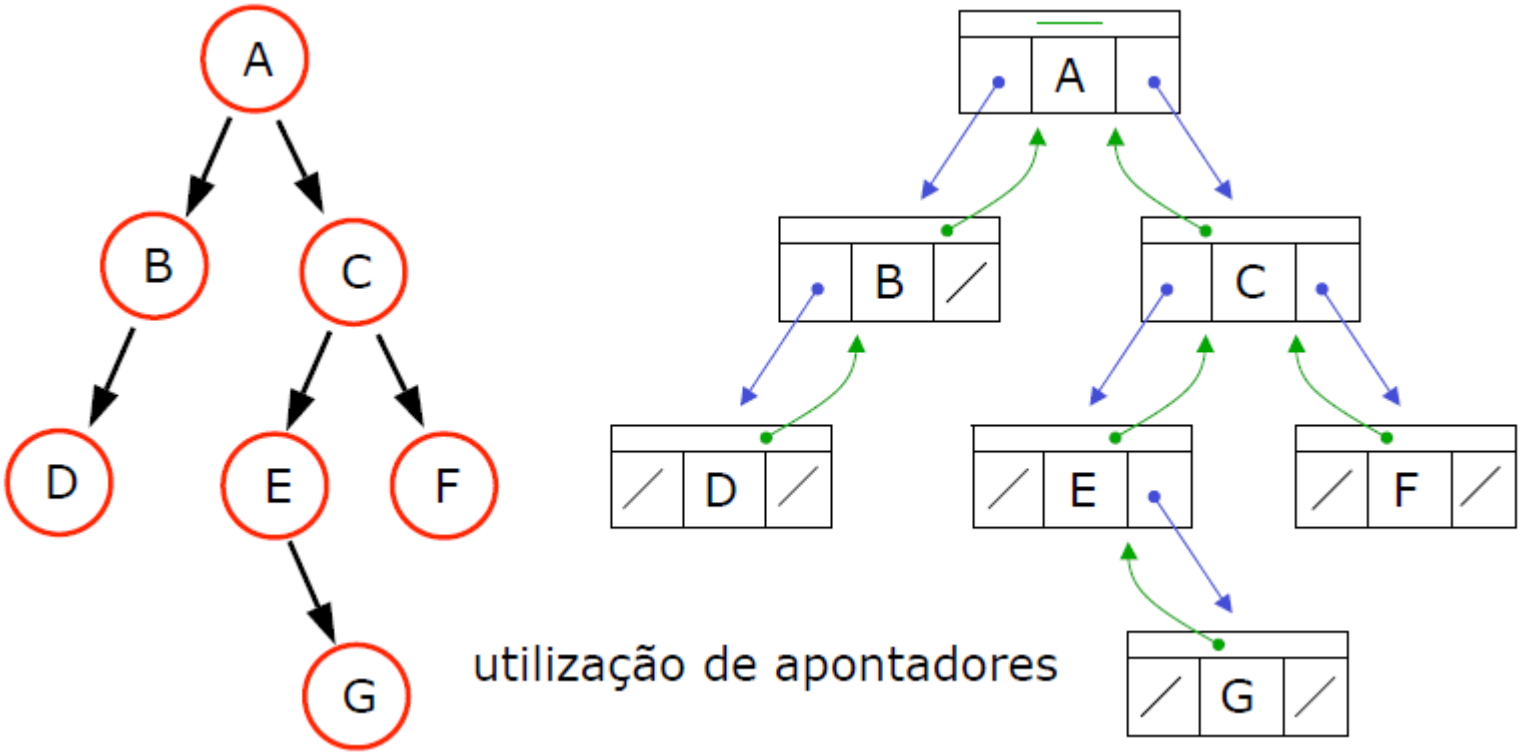
- Estrutura:

```
typedef struct cel {
   int chave;
   struct cel *esq;
   struct cel *dir;
} no;

typedef no arvore;
```

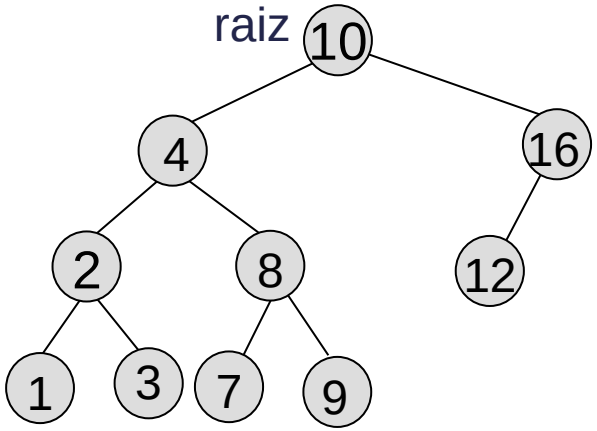


Representação da árvore binária

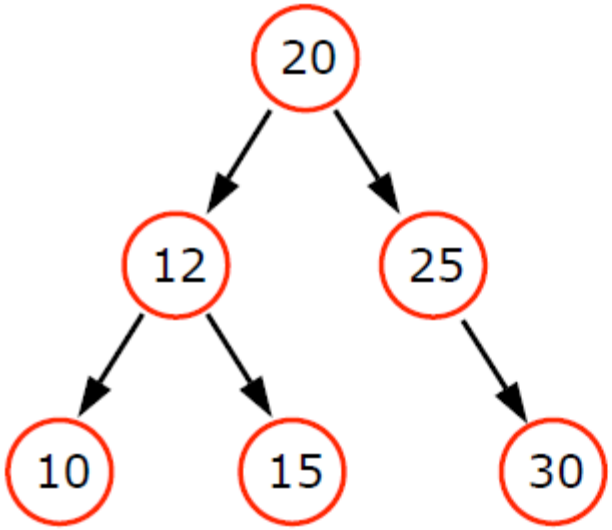


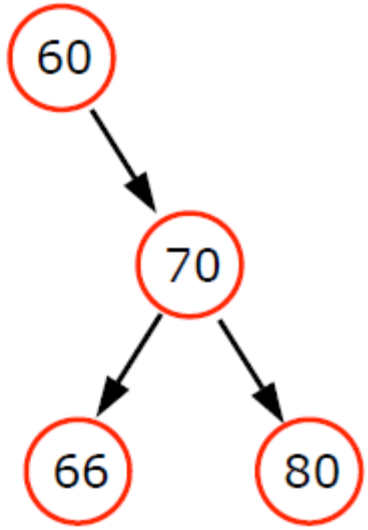
Árvore binária de pesquisa (ou busca)

- Propriedades de uma árvore binária de pesquisa :
  - cada chave do nó X é maior ou igual à chave de qualquer nó na subárvore **esquerda** de X
  - cada chave do nó X é menor ou igual à chave de qualquer nó na subárvore **direita** de X
- Dessa forma, a varredura da árvore em ordem e-r-d vê as chaves em ordem crescente.

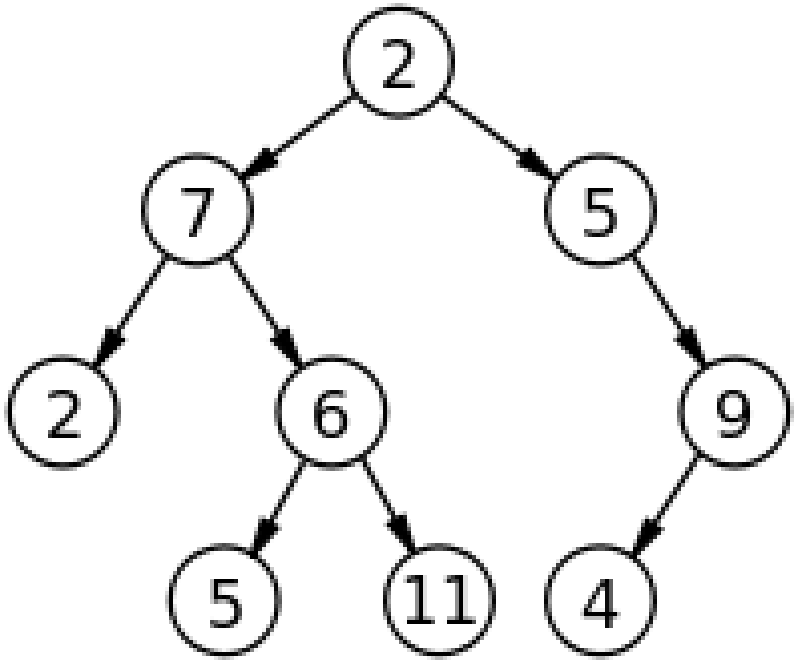


Exemplos



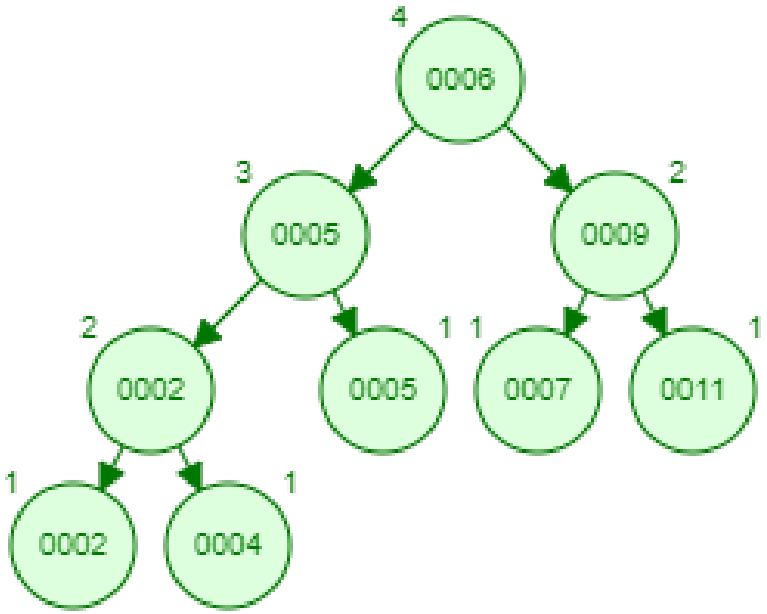


Balanceamento de árvore



A árvore acima não está balanceada (elemento 5 possui 2 filhos a direita e nenhum a esquerda), nem está ordenada. Faça a correção tornando a árvore balanceada e ordenada.

Balanceamento de árvore



Agora a árvore está balanceada e ordenada

Percursos em Árvore Binária Balanceada

Percorrer uma árvore binária em **pré-ordem**:

1 - Visitar a raiz.

2 - Percorrer a sua subárvore esquerda em pré-ordem.

3 - Percorrer a sua subárvore direita em pré-ordem.

Percursos em Árvore Binária Balanceada

Percorrer uma árvore binária em **in-ordem**:

1 - Percorrer a sua subárvore esquerda em in-ordem.

2 - Visitar a raiz.

3 - Percorrer a sua subárvore direita em in-ordem

Percursos em Árvore Binária Balanceada

Percorrer uma árvore binária em **pós-ordem**:

1 - Percorrer a sua subárvore esquerda em pós-ordem.

2 - Percorrer a sua subárvore direita em pós-ordem.

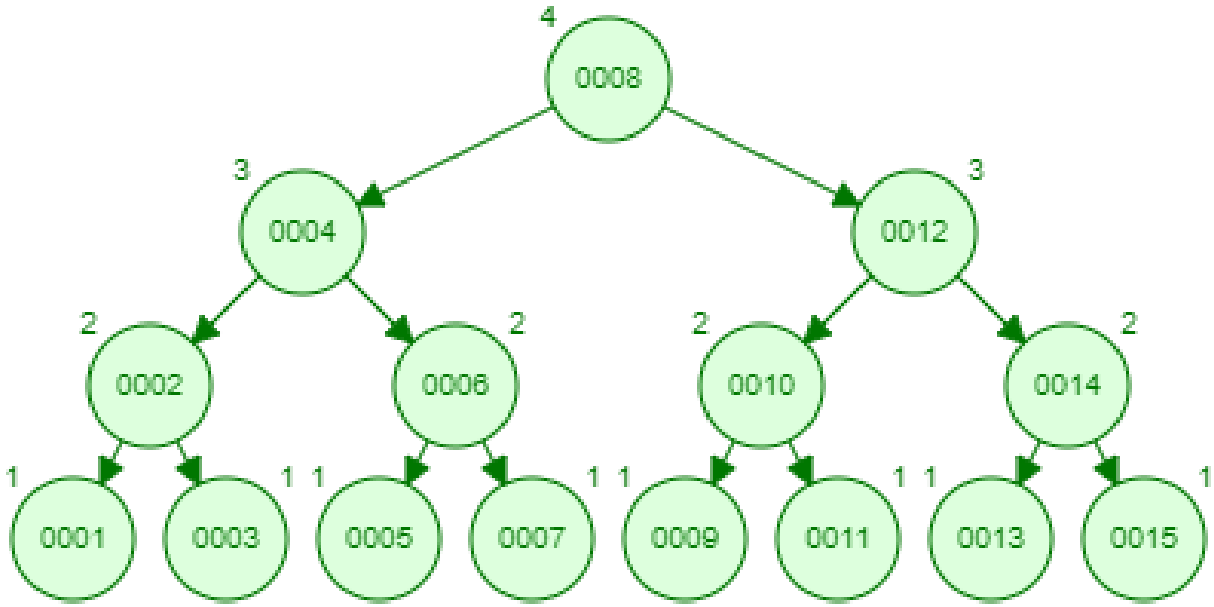
3 - Visitar a raiz.

Exercício

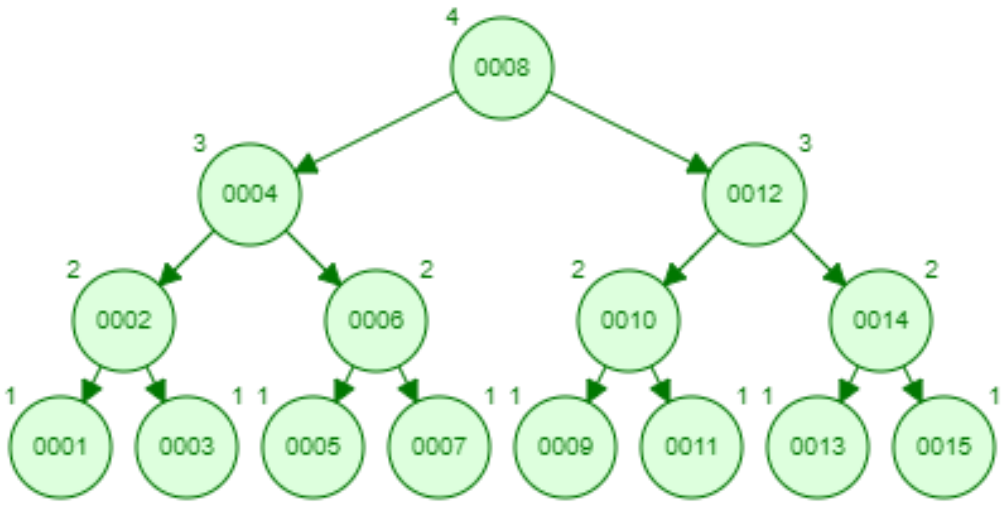
Desenhe uma árvore binária balanceada (AVL) com os elementos abaixo. Após a representação, exiba os itens em percurso pré-ordem, in-ordem e pós-ordem.

1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 -12 – 13 – 14 - 15

Resposta

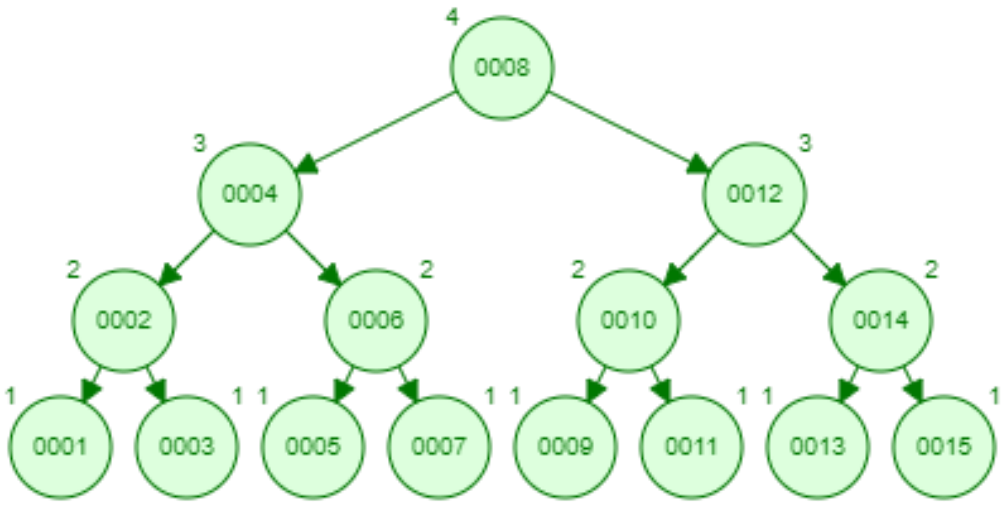

4
0008
3
0004
3
0012
2
0002
2
0006
2
0010
2
0014
1
0001
1
0003
1
0005
1
0007
1
0009
1
0011
1
0013
1
0015

Resposta



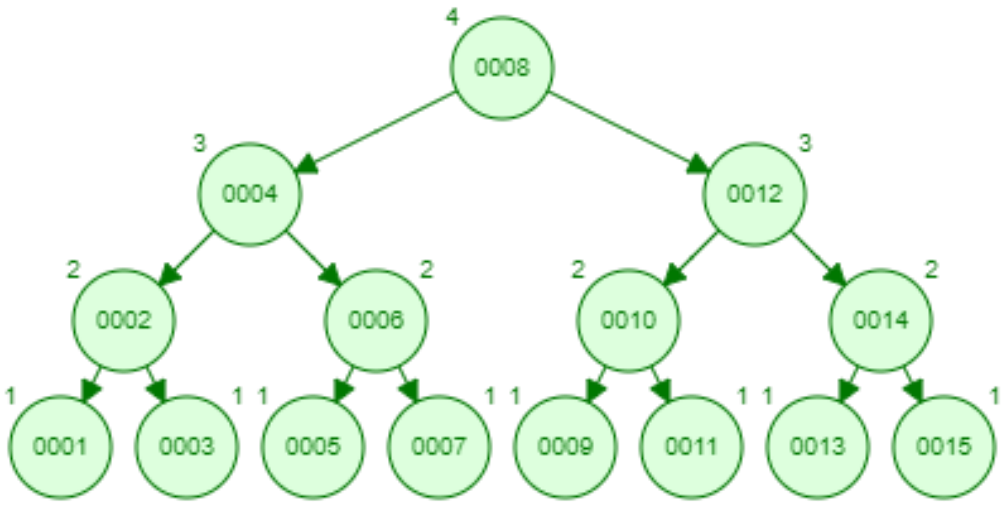
Pré-Ordem: 8 – 4 – 2 – 1 – 3 – 6 – 5 – 7 – 12 – 10 – 9 – 11 – 14 – 13 - 15

Resposta



In-Ordem: 1 – 2 – 3 – 4 - 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 - 14 -15

Resposta



Pós-Ordem: 1 – 3 – 2 – 5 – 7 – 6 – 4 – 9 – 11 – 10 – 13 – 15 - 14 – 12 - 8

Referências

- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Makron books.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto Castilho. *Algoritmos e estruturas de dados*. LTC Editora.
- FIDALGO, Robson. *Material para aulas*. UFRPE.
- LOPES, Anita; GARCIA, Guto. *Introdução à programação – 500 algoritmos resolvidos*. Elsevier.
- NELSON, Fábio. *Material para aulas: Algoritmo e Programação.* UNIVASP.
- FEOFILOFF, P., *Algoritmos em linguagem C*, Editora Campus, 2008.
- ZIVIANI, N., *Projeto de algoritmos com Implementações em Pascal e C*, São Paulo: Pioneira, 2d, 2004.
- http://www.ime.usp.br/~pf/algoritmos/
- SANTOS, Wellington Lima. *Material para aulas: Fila (Queue)*, UFMS/CPDO/DEX.