

AULA e07a

Algoritmos e Estruturas de Dados I

Árvores Binárias - Percursos (parte I)

Luciano Antonio Digiampietri

Árvores Binárias - Percursos

Há diversas formas de percorrer uma árvore.

Árvores Binárias - Percursos

Há diversas formas de percorrer uma árvore.

- Utilizamos **abordagens recursivas** para exibir os elementos de uma árvore, contar o número de nós ou calcular sua altura.

Árvores Binárias - Percursos

Há diversas formas de percorrer uma árvore.

- Utilizamos **abordagens recursivas** para exibir os elementos de uma árvore, contar o número de nós ou calcular sua altura.
- Hoje discutiremos **diferentes abordagens**.

Árvores Binárias - Percursos

Os percursos mais comuns em árvores binárias partem da **raiz** e, recursivamente, percorrem a **subárvore à esquerda** e em seguida a **subárvore à direita**.

Árvores Binárias - Percursos

Os percursos mais comuns em árvores binárias partem da **raiz** e, recursivamente, percorrem a **subárvore à esquerda** e em seguida a **subárvore à direita**.

Estes percursos podem ser implementados de maneira **iterativa** (utilizando uma estrutura de dados auxiliar);

Árvores Binárias - Percursos

Os percursos mais comuns em árvores binárias partem da **raiz** e, recursivamente, percorrem a **subárvore à esquerda** e em seguida a **subárvore à direita**.

Estes percursos podem ser implementados de maneira **iterativa** (utilizando uma estrutura de dados auxiliar);

É possível também percorre a árvore **em nível**.

Árvores Binárias

Discutiremos as seguintes funções:

Exibição dos elementos em *pré-ordem*, *em ordem* e em *pós-ordem* (implementações recursivas);

Exibição dos elementos em *pré-ordem* (implementação iterativa);

Exibição dos elementos *em nível* (implementação iterativa);

Modelagem

```
#include <stdio.h>
#include <malloc.h>
#define true 1
#define false 0
```

```
typedef int bool;
typedef int TIPOCHAVE;
```

```
typedef struct aux{
    TIPOCHAVE chave;
    struct aux *esq, *dir;
} NO;
```

```
typedef NO* PONT;
```

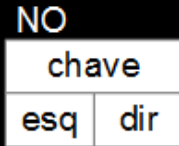
Modelagem

```
#include <stdio.h>
#include <malloc.h>
#define true 1
#define false 0

typedef int bool;
typedef int TIPOCHAVE;

typedef struct aux{
    TIPOCHAVE chave;
    struct aux *esq, *dir;
} NO;

typedef NO* PONT;
```



Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
}
```

Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

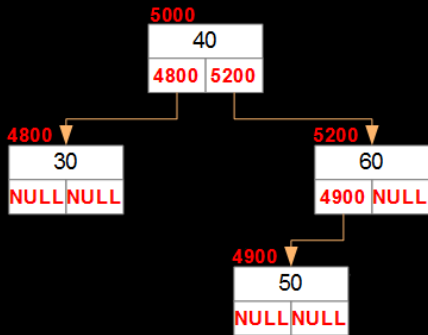
```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

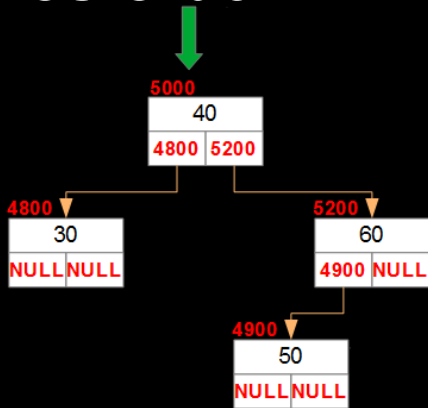


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

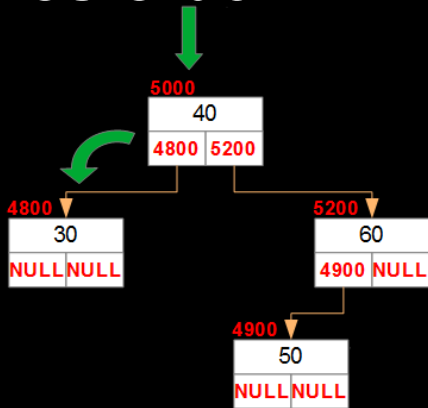


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

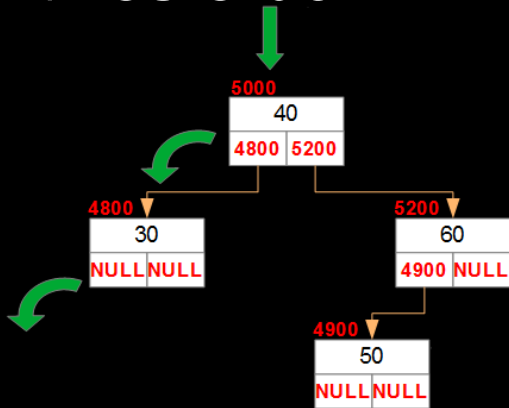


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

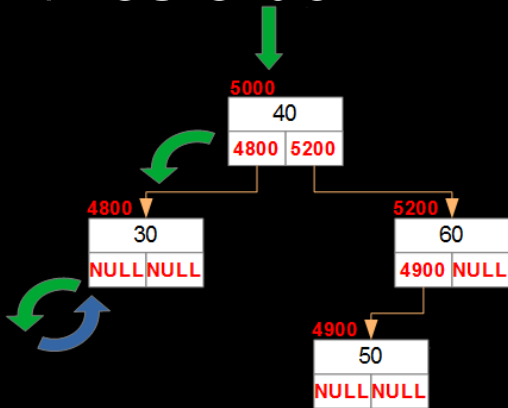


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}    $ 30
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

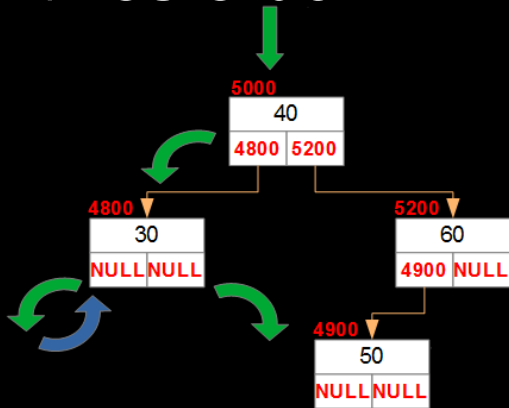


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}    $ 30
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}
```

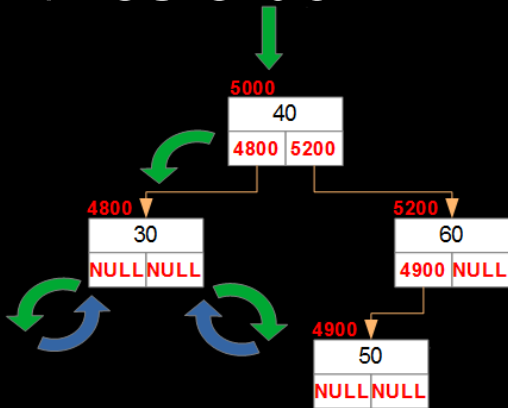


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30
```

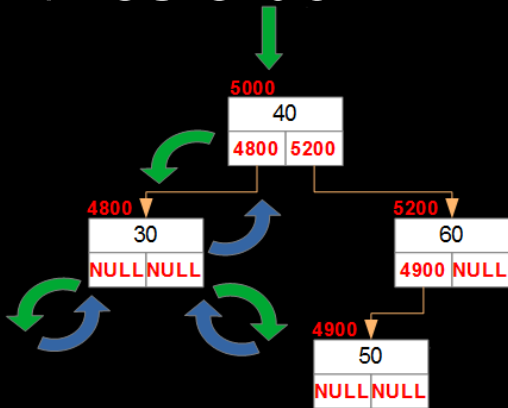


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30
```

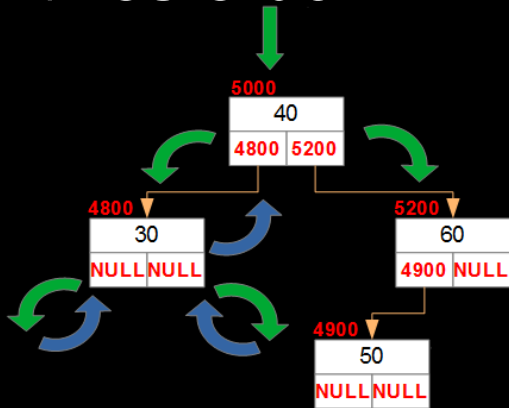


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30
```

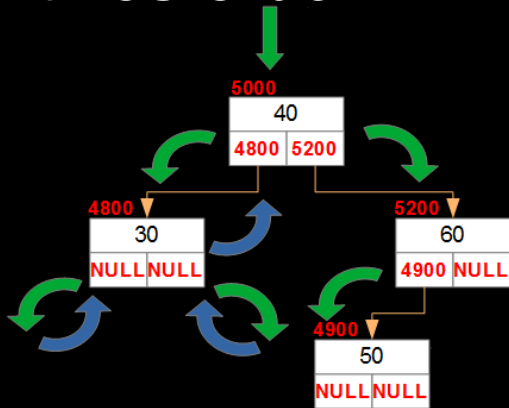


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30
```

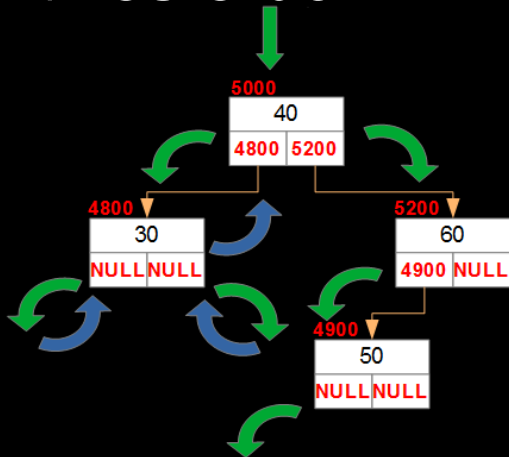


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}    $ 30 40
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}    $ 30
```

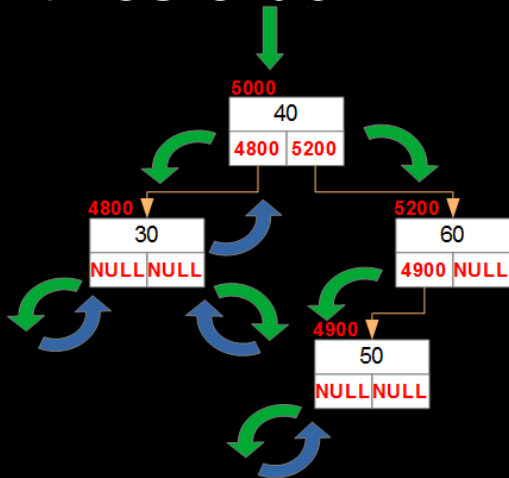


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30
```

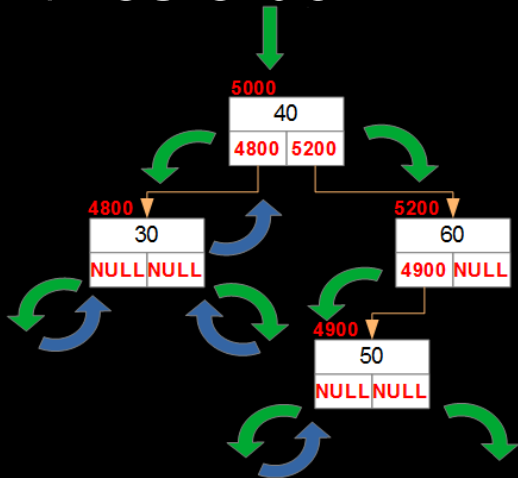


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}    $ 30 40 50
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}    $ 30
```

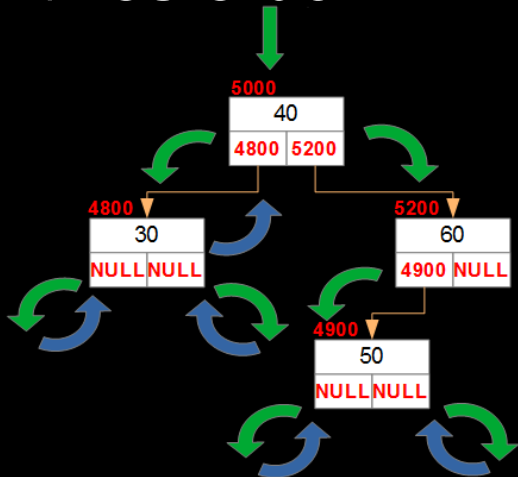


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}    $ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}    $ 30 40 50
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}    $ 30 50
```

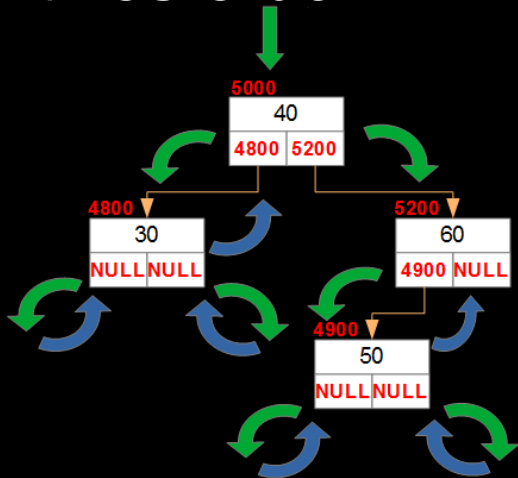


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50 60
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30 50
```

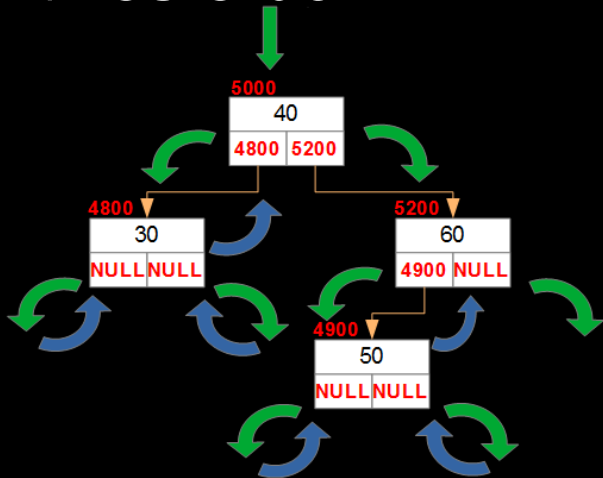


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50 60
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30 50
```

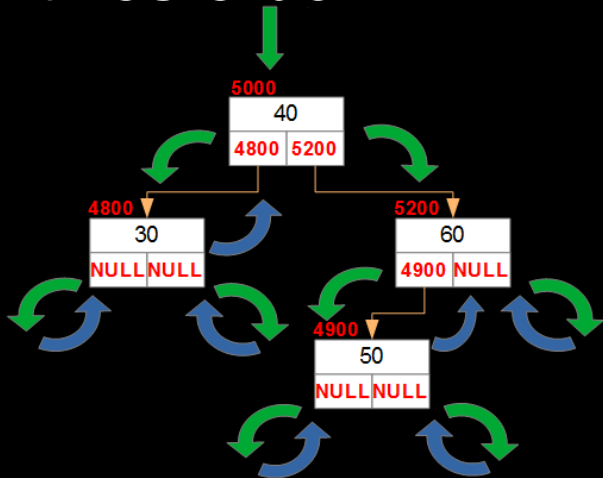


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50 60
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30 50 60 40
```

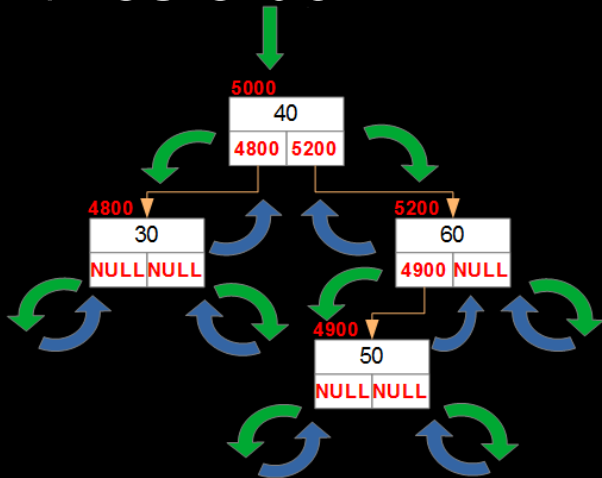


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50 60
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30 50 60 40
```

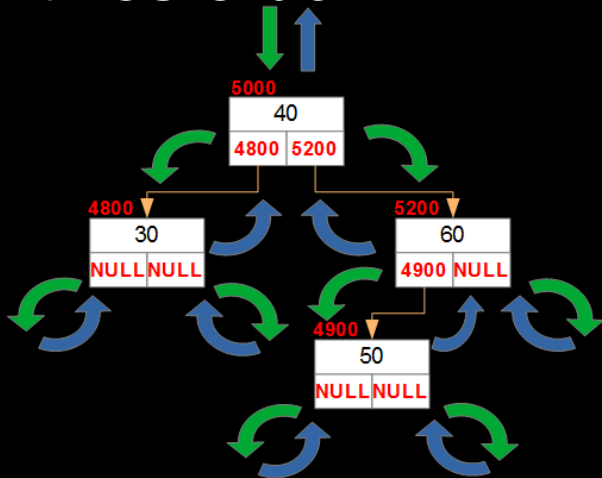


Percursos Pré/Em/Pós-ordem

```
void exibirArvorePreOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    printf("%i ",raiz->chave);  
    exibirArvorePreOrdem(raiz->esq);  
    exibirArvorePreOrdem(raiz->dir);  
}  
$ 40 30 60 50
```

```
void exibirArvoreEmOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvoreEmOrdem(raiz->esq);  
    printf("%i ",raiz->chave);  
    exibirArvoreEmOrdem(raiz->dir);  
}  
$ 30 40 50 60
```

```
void exibirArvorePosOrdem(PONT raiz){  
    if (raiz == NULL) return;  
    exibirArvorePosOrdem(raiz->esq);  
    exibirArvorePosOrdem(raiz->dir);  
    printf("%i ",raiz->chave);  
}  
$ 30 50 60 40
```



AULA e07a

Algoritmos e Estruturas de Dados I

Árvores Binárias - Percursos (parte I)

Luciano Antonio Digiampietri