

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PIAUÍ
Campus Teresina - Central

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO PIAUÍ

CAMPUS TERESINA-CENTRAL

DIRETORIA DE ENSINO

Estrutura de Dados II – Percurso em Profundidade Em Árvores Alinhadas - Aula 4 -

Professora: Elanne Cristina O. dos Santos

elannecristina.santos@gmail.com

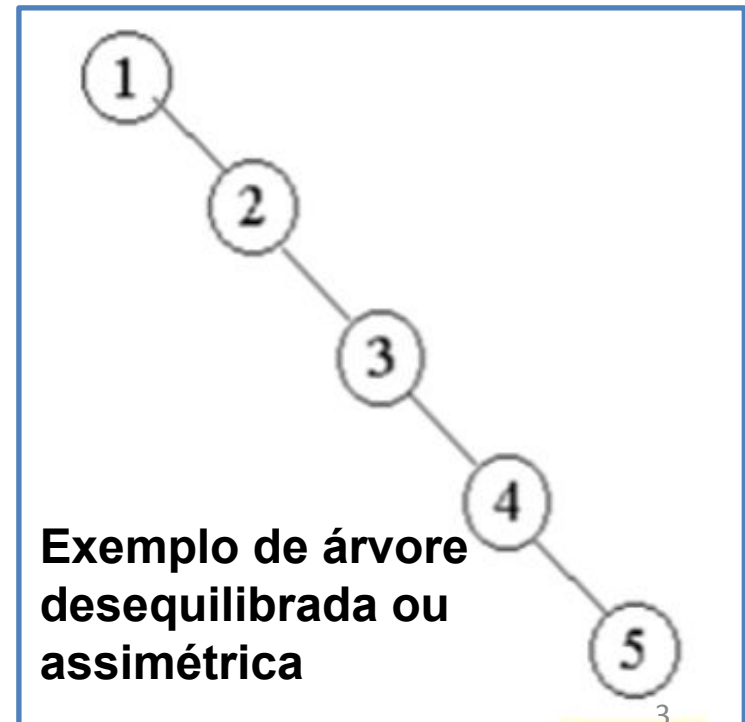
elannecristina.santos@ifpi.edu.br

Árvores Alinhadas

- Segundo o autor:
 - “As funções de percurso analisadas anteriormente foram tanto recursivas como não recursivas, mas ambos os tipos usaram uma pilha tanto implícita como explicitamente.”
 - “No caso das funções recursivas, a pilha em tempo de execução foi utilizada.
 - “No caso das funções não-recursivas, uma pilha explicitamente definida e mantida pelo usuário foi usada.”

Árvores Alinhadas

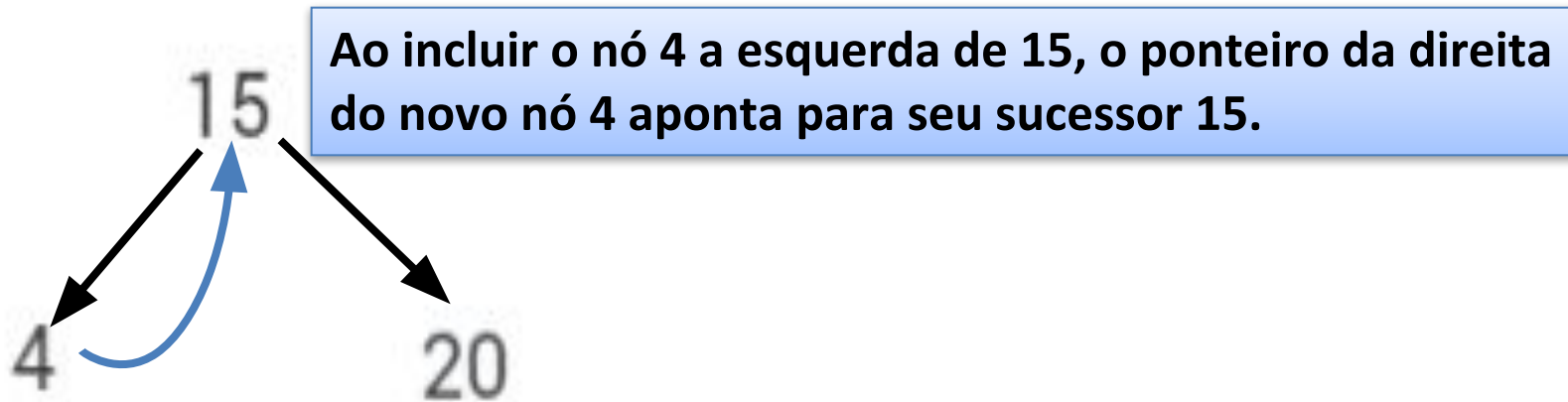
- Segundo o autor:
 - “A questão é que algum tempo adicional tem que ser gasto para manter a pilha, e mais algum espaço deve ser gasto para a própria árvore. “
 - “No pior caso, quando a árvore é desequilibrada
é desequilibrada
desfavoravelmente, a pilha
pode conter informação
sobre quase todos os nós
da árvore, uma séria
questão para árvores
muito grandes.”



Árvores Alinhadas

- Proposta:
- O ponteiro da direita de um nó da árvore aponta para um filho ou aponta para um sucessor;
- Aonde o sucessor se entende pelo nó pai do nó atual.

Ex.:

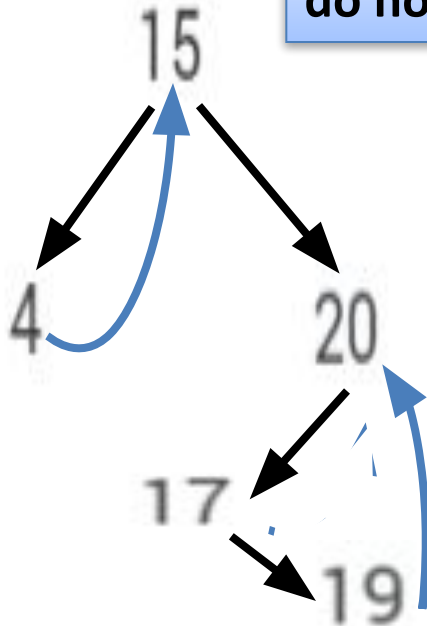


Ao incluir o nó 20 a direita ele verifica se 15 tem sucessor, como 15 não tem sucessor ele simplesmente inclui o novo nó 20 a direita da árvore.

Árvores Alinhadas

- Assim:
 - Um nó com um filho a direita tem um sucessor em algum lugar na sua subárvore à direita ou
 - Um nó sem filho a direita tem seu sucessor em algum lugar acima dele.
 - Cont. Ex:

Ao incluir o nó 17 a esquerda de 20, o ponteiro da direita do novo nó 17 aponta para seu sucessor.



Ao incluir o nó 19 a direita do nó 17, o novo nó 19 recebe o sucesso de 17. O nó 17 deixa de ter um sucessor.

Árvores Alinhadas

- Exceto para o nó raiz, todos os nós sem filhos a direita terão linhas para os seus sucessores.
- Se um nó se torna filho a direita do outro, ele herda o sucessor do nó pai.

Implementando a árvore alinhada

- Primeiro passo: Mudar a estrutura do nó da árvore:
 - Como um ponteiro pode apontar para um nó de cada vez, é necessário controlar se este ponteiro está apontando para um filho ou para um sucessor.
 - Sugestão: ter um campo a mais na árvore chamado sucessor que marque **True** ou **False** (**1** ou **0**) para cada uma das opções.

```

class ArvoreNo {
    public:
        T el;
        ArvoreNo<T> *left,*right;
        int sucessor;
        ArvoreNo(){
            left=right=0;
            sucessor=0;
        }
        ArvoreNo(T e,ArvoreNo<T> *l=0,ArvoreNo<T> *r=0,int s=0){
            el=e;
            left=l;
            right=r;
            sucessor=s;
        }
};

```

**OBS: implementação completa
no livro pag. 210 – figura 6.24**


```

if (root == 0){
    root = newNode;
    return;
}
p = root;
while (p!=0){
    prev = p;
    if(p->el>el)
        p=p->left;
    else if (p->sucessor==0)
        p=p->right;
    else break;
}
if (prev->el>el){
    prev->left=newNode;
    newNode->sucessor=1;
    newNode->right=prev;
}
else if (prev->sucessor==1){
    newNode->sucessor = 1;
    prev->sucessor=0;
    newNode->right=prev->right;
    prev->right=newNode;
}
else prev->right=newNode;

```

Atividade

- Implementar inclusão de uma árvore binária alinhada.
- Implemente o percurso pré-ordem na árvore alinhada.
- Verifique o tempo de execução de cada um dos algoritmos nas três versões (árvore binária com recursão, árvore binária sem recursão com pilha e árvore alinhada) , confira se realmente o uso da estrutura alinhada otimizou o algoritmo gastando menos tempo de processamento.
 - Existe melhoria de eficiência do algoritmo?