

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 – Ordem de percurso - Aula 2 -

Professora: Elanne Cristina O. dos Santos

elannecristina.santos@gmail.com

elannecristina.santos@ifpi.edu.br

Percurso em profundidade

- pré-ordem (VLR) , in-ordem (LVR) e pós-ordem (LRV).
- Exemplo (usando recursão):

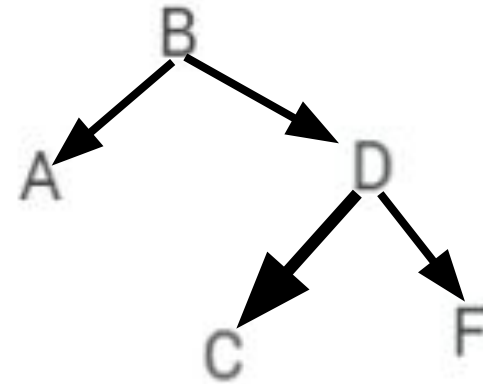
```
void preorder(ArvoreNo<T> *p){  
    if (p!=0) {  
        visit(p);  
        preorder(p->left);  
        preorder(p->right);  
    }  
}
```

Percurso em Profundidade

- Usando recursão:
 - “Funções simples, mas com seu real poder na recursão dupla. O trabalho é feito pelo sistema na pilha em tempo de execução. Simplifica a codificação, mas coloca uma pesada carga sobre o sistema. “
 - “Uma implementação recursiva tem a tendência de ser menos eficiente do que uma não recursiva.”
 - “Considere o problema de uma implementação não recursiva dos 3 algoritmos.”

Pré-ordem : Solução sem recursão /usando pilha

- Push B

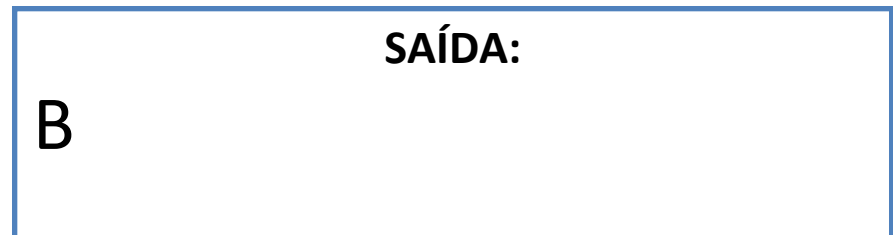
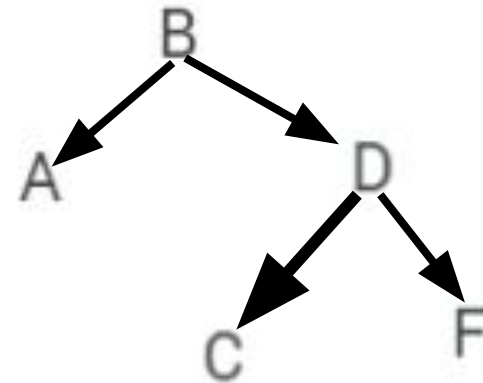
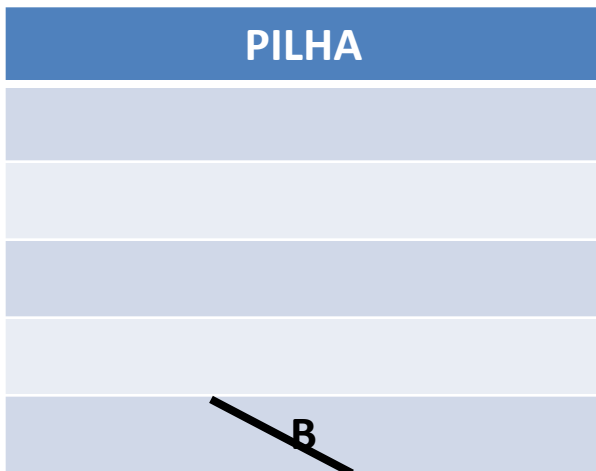


PILHA
B

SAÍDA:

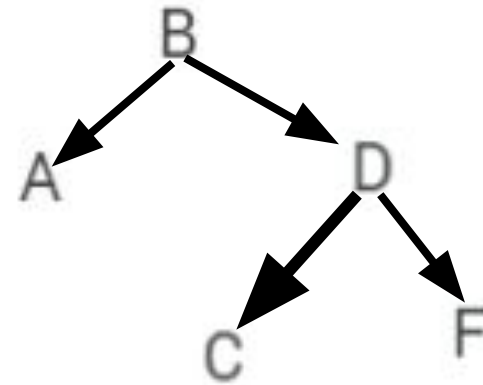
Pré-ordem : Solução sem recursão /usando pilha

1. Pop B
2. Imprime B



Pré-ordem : Solução sem recursão /usando pilha

1. Push D
2. Push A



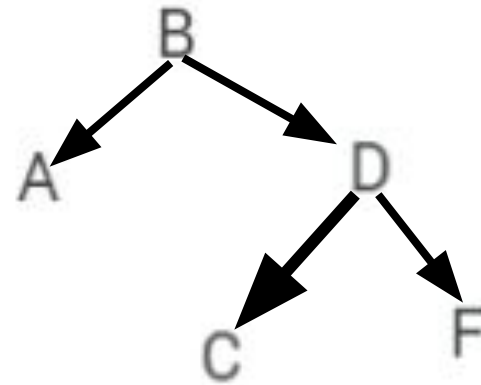
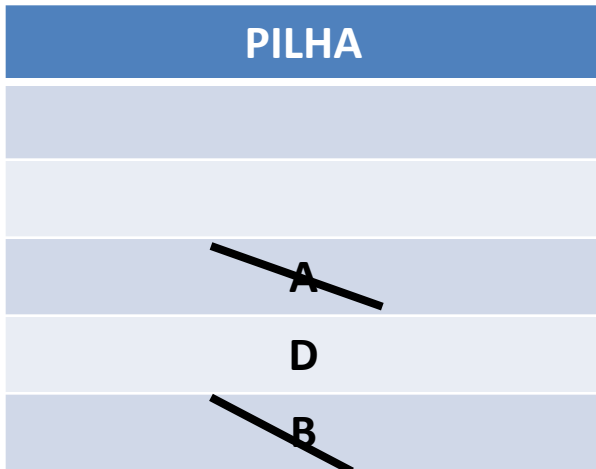
PILHA
A
D
B

SAÍDA:

B

Pré-ordem : Solução sem recursão /usando pilha

1. Pop A
2. Imprime A:

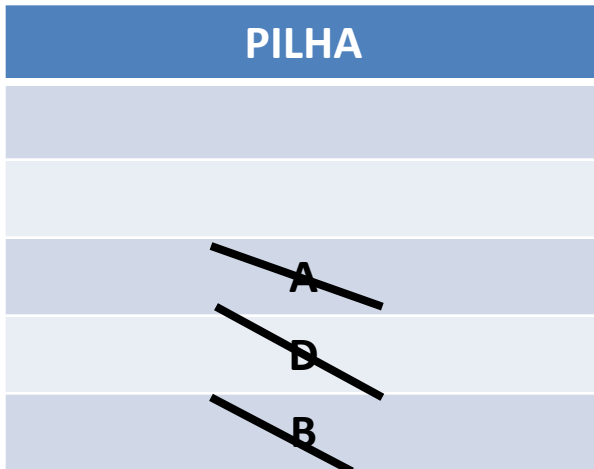
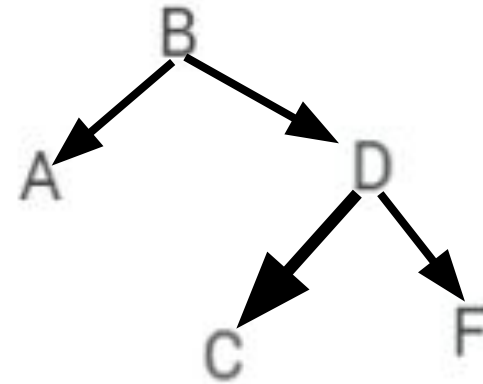


SAÍDA:

B - A

Pré-ordem : Solução sem recursão /usando pilha

1. Pop D
2. Imprime D:

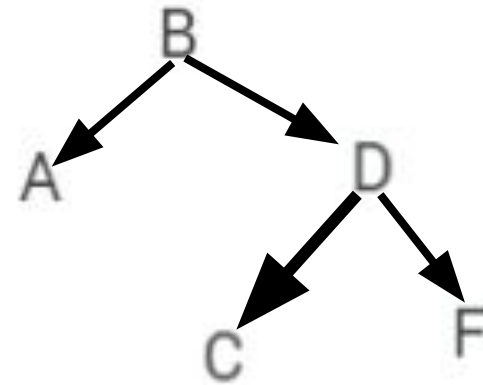


SAÍDA:

B – A - D

Pré-ordem : Solução sem recursão /usando pilha

1. Push F
2. Push C



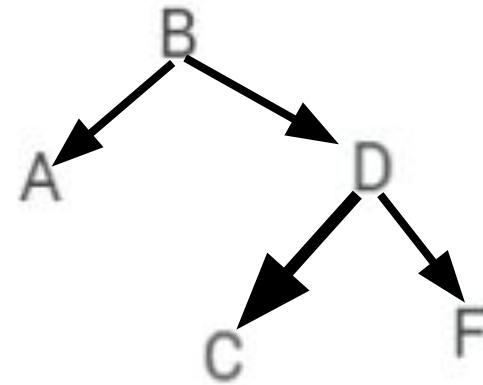
PILHA
C
F
A
D
B

SAÍDA:

B – A - D

Pré-ordem : Solução sem recursão /usando pilha

1. Pop C
2. Imprime C



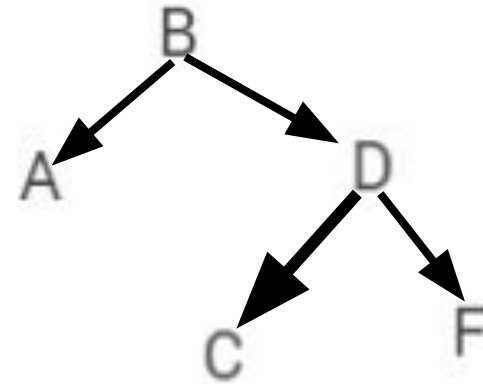
PILHA
C
F
A
D
B

SAÍDA:

B – A – D – C

Pré-ordem : Solução sem recursão /usando pilha

1. Pop F
2. Imprime F:



PILHA
C
F
A
D
B

SAÍDA:

B - A - D - C - F

Solução sem recursão /usando pilha (fig. 6.15 pag 201):

```
void preOrder(){
    stack<No*> pilha;
    No *p = raiz;
    string v;
    if (p!=0){
        pilha.push(p);
        while (!pilha.empty()) {
            p=pilha.top();
            cout<<pilha.top()->nome<<endl;
            pilha.pop();
            if (p->right !=0)
                pilha.push(p->right);
            if (p->left != 0)
                pilha.push(p->left);
        }
    }
}
```