

Vinícius do Amaral Brunheroto
Estrutura de dados I
Profº Carlos Fischer
Unesp RC

ATIVIDADE 2

A)

```
/* declarar árvore usando ponteiro */
```

```
tipo NoPtr = *No;
```

```
    No = estrutura
```

```
        info : char;
```

```
        esq, dir, ant : NoPtr;
```

```
    fim;
```

```
var Tree : NoPtr;
```

```
/* subrotina para percorrer a árvore */
```

```
função PrePerc (t: NoPtr);
```

```
flag1: int /* flag1: serve como auxiliar para contar se percorreu a raiz e nós à esquerda da árvore */
```

```
flag2: int /* flag2: serve para contar se percorreu a raiz e nós à direita da árvore */
```

```
flag3: int /* flag3: serve para informar se já foi printado o elemento do nó que é pai de nó folha */
```

```
r: NoPtr /* serve para os casos em que há nós únicos no lado direito da árvore */
```

```
    começo
```

```
    se (t<> null) /*comparação */
```

```
        começo
```

```
        /* visitar a raiz */
```

```
        /* PERCORRER UM NÓ (raiz ou à esquerda)*/
```

```
        flag1++
```

```
        PrePerc (t->esq); /* percorrer s.a. esquerda em PrePerc */
```

```
        r = t->ant /* aponta para o nó anterior a t */
```

```
        flag3=0
```

```
        se (flag1 >= 2) /* percorreu no mínimo a raiz e um nó a esquerda */
```

```
            {
```

```
                escrever (t → info);
```

```
                flag3 = 1 /* para informar que já escreveu um valor na tela */
```

```
}
```

```
flag1 =0
```

```
/* PERCORRER UM NÓ (raiz ou à direita)*/  
flag2++
```

```
PrePerc (t->dir); /* percorrer s.a. direita em PrePerc */
```

```
se( (flag2 >= 2) e (flag3 != 1)) /* percorreu no mínimo a raiz e um nó a direita */  
    escrever (r); /* escrever conteúdo do ponteiro r (com endereço anterior à t, dessa  
forma aponta-se para o pai do nó folha) */
```

```
flag2 =0  
flag3 =0
```

```
fim;
```

```
fim;
```

chamada do algoritmo desenvolvido:

```
PrePerc (NoPtr);
```

B) O algoritmo feito faz comparações em cada nó (se o ponteiro aponta pra NULL).
O algoritmo tende a percorrer todos os nós da árvore, uma vez para cada nó, para poder verificar qual nó é considerado pai de um nó folha. Por isso, a complexidade/eficiência é $O(n)$.