

UNIVERSIDADE FEDERAL DO CEARÁ

ESTRUTURAS DE DADOS - TIO140 2022.2 T02A

AULA 21 - 31/10/2022 - Segunda-feira

IMPLEMENTAÇÃO DE ÁRVORE BINÁRIA DE BUSCA

//CONTINUANDO:

```
STRUCT Noh { int elem; Noh *esq, *dir; };
```

```
STRUCT ArvBB
```

```
{
```

```
  Noh *raiz;
```

```
  ArvBB(): raiz(NULLPTR) { }
```

```
  void inserir(int e) { ... }
```

```
}
```

1 2

STRUCT ArvBB

bool pertence(int e)

```
{
```

```
  Noh *n = raiz;
```

```
  while (n != NULLPTR)
```

```
  {
```

```
    if (e == n->elem) return true;
```

```
    if (e < n->elem) n = n->esq;
```

```
    else
```

```
      n = n->dir;
```

```
  }
```

```
  return false;
```

```
}
```

void remover(int e)

```
{
```

```
  Noh *n = raiz, *cima = NULLPTR;
```

```
  while (n != NULLPTR)
```

```
  {
```

```
    if (e == n->elem) break;
```

```
    if (e < n->elem) { cima = n; n = n->esq; }
```

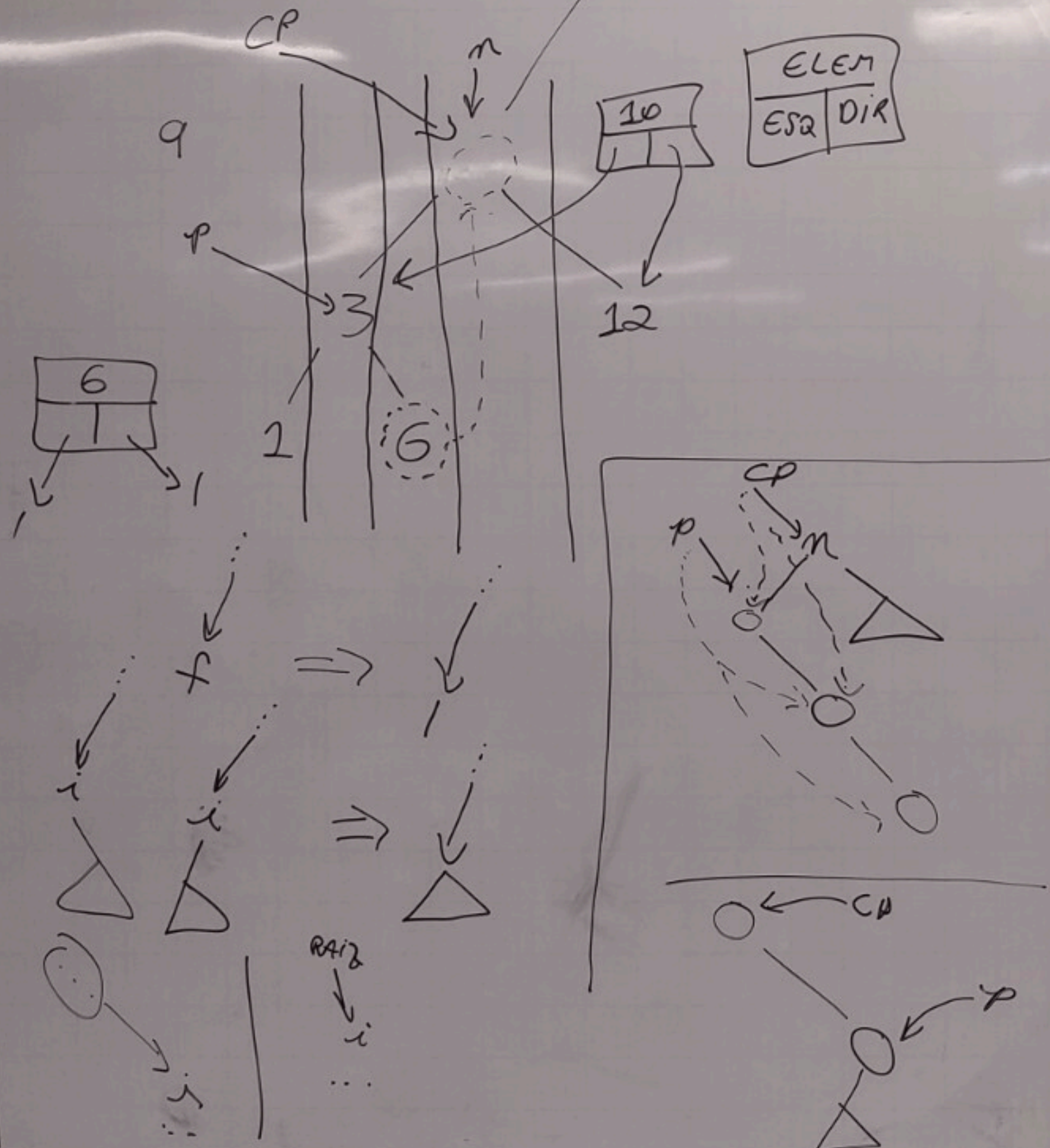
```
    else
```

```
      { cima = n; n = n->dir; }
```

```
  }
```

2 3

2 3



ArvBB REMOVER

```

if (n == NULLPTR) RETURN; // NÃO ENCONTRADO
if (n->ESQ == NULLPTR)
{
    if (CIMA == NULLPTR) RAIZ = n->DIR;
    ELSE if (n->ELEM < CIMA->ELEM) CIMA->ESQ = n->DIR;
    ELSE /* n == CIMA->DIR */ CIMA->DIR = n->DIR;
}
ELSE if (n->DIR == NULLPTR)
{
    if (CIMA == NULLPTR) RAIZ = n->ESQ;
    ELSE if (n == CIMA->ESQ) CIMA->ESQ = n->ESQ;
    ELSE CIMA->DIR = n->ESQ;
}

```


5/ ArvBB REMOVER

ELSE
{

Noh $*p = n \rightarrow \text{ESQ}$, $*cp = n$;

WHILE ($p \rightarrow \text{DIR} \neq \text{NULLPTR}$)

{

$cp = p$;

$p = p \rightarrow \text{DIR}$;

}

if ($p == cp \rightarrow \text{ESQ}$) $cp \rightarrow \text{ESQ} = p \rightarrow \text{ESQ}$;

ELSE $/ * p == cp \rightarrow \text{DIR} */$ $cp \rightarrow \text{DIR} = p \rightarrow \text{ESQ}$;

$p \rightarrow \text{ESQ} = n \rightarrow \text{ESQ}$;

$p \rightarrow \text{DIR} = n \rightarrow \text{DIR}$;

if ($\text{cima} == \text{NULLPTR}$) $\text{RAIZ} = p$;

ELSE if ($n == \text{cima} \rightarrow \text{ESQ}$) $\text{cima} \rightarrow \text{ESQ} = p$;

ELSE $\text{cima} \rightarrow \text{DIR} = p$;

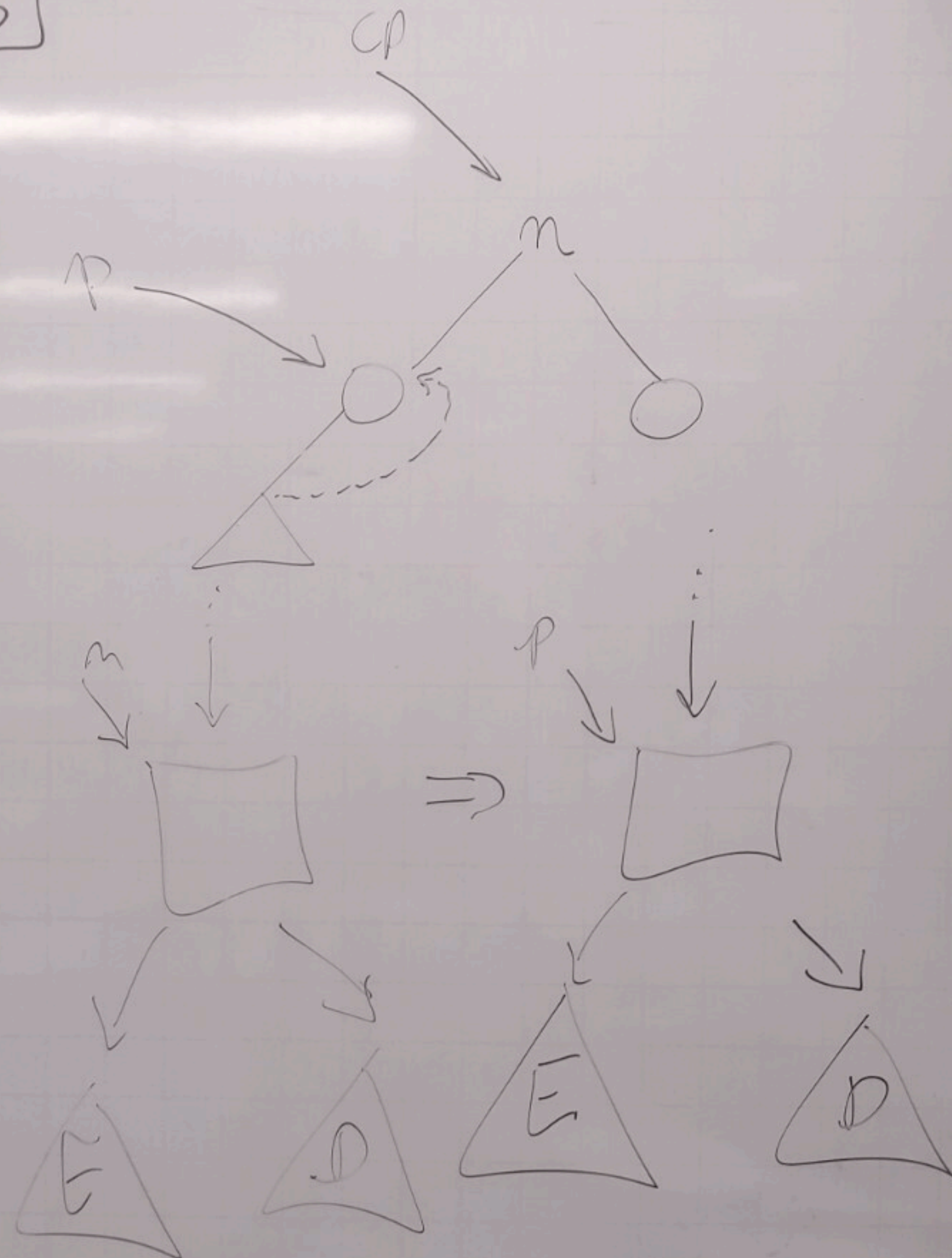
ELSE
{

DELETE n ;

} // REMOVER

DESALOCAR n

5/6



Arv BB

VOID REMOVER_MELHOR (int E)

{

Noh $**CIMA = \&RAIZ$; Noh $*n$;

for(;;) // WHILE(TRUE)

{

$n = *CIMA$; // O PONTEIRO APONTADO POR "CIMA"

if ($n == NULLPTR$) RETURN;

if ($E == n \rightarrow elem$) BREAK;

if ($E < n \rightarrow elem$) $CIMA = \&n \rightarrow ESQ$;

ELSE

$CIMA = \&n \rightarrow DIR$;

}

if ($n \rightarrow ESQ == NULLPTR$)

{ $*CIMA = n \rightarrow DIR$; }

ELSE if ...

}

7 8

