

UNIVERSIDADE FEDERAL DO CEARÁ
CK0109 - 2019.2 - T02 - ESTRUT. DADOS
AULA 14 - 11/09/2019

ITERADORES (CONT.)

1. CONTINUAÇÃO DA IMPLEMENTAÇÃO DE CONJUNTO:

// CONJUNTO

ITERADOR inicio ()
{

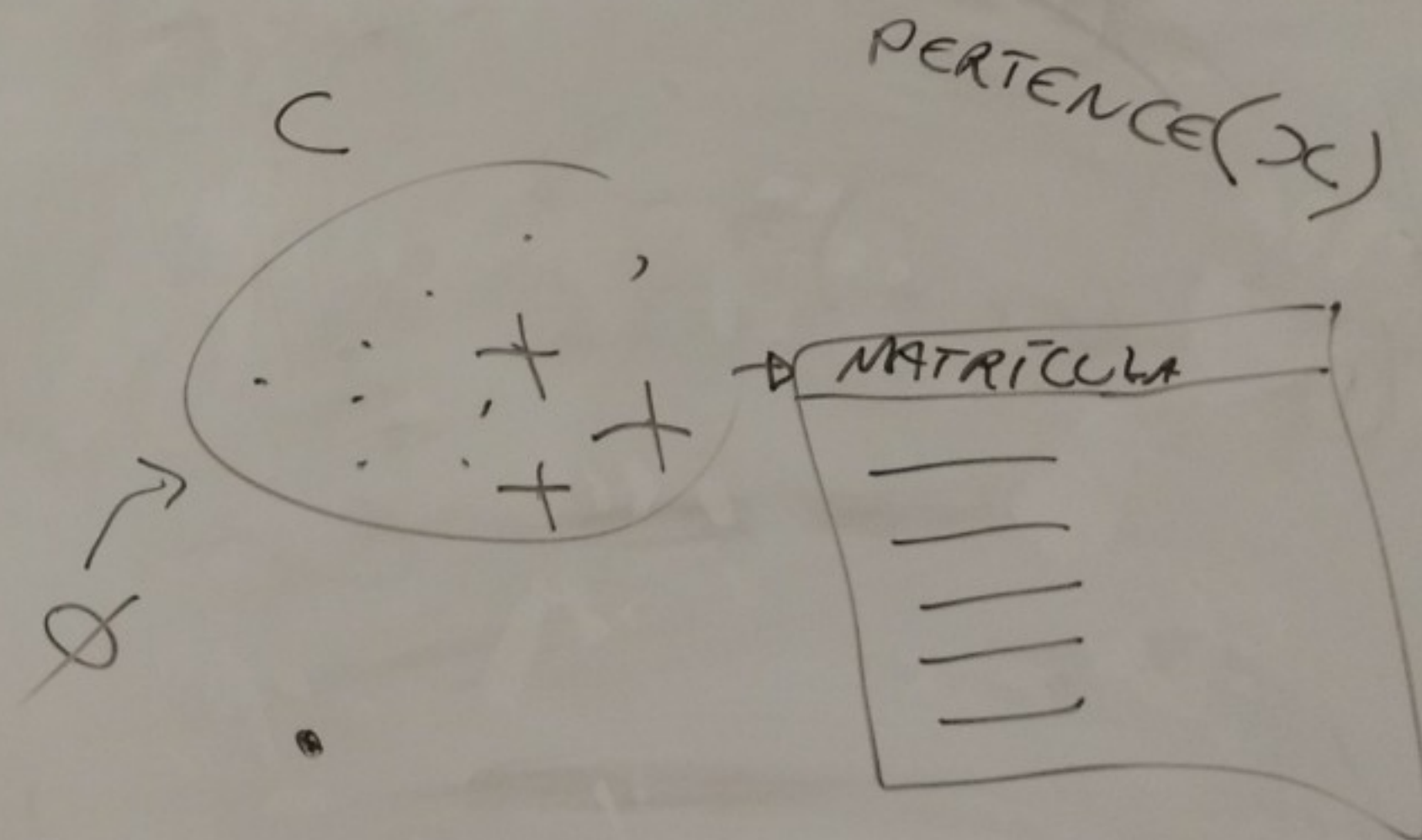
ITERADOR i; i.p = &(v[0]);
RETURN i;
}

// CONJUNTO

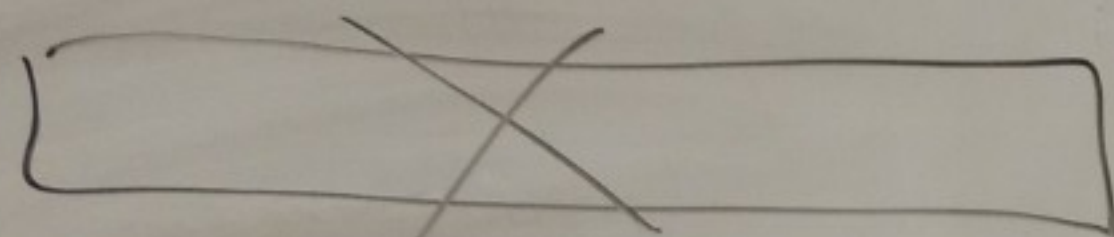
ITERADOR fim ()

{
ITERADOR i; i.p = &(v[ULT+1]);
RETURN i;
}

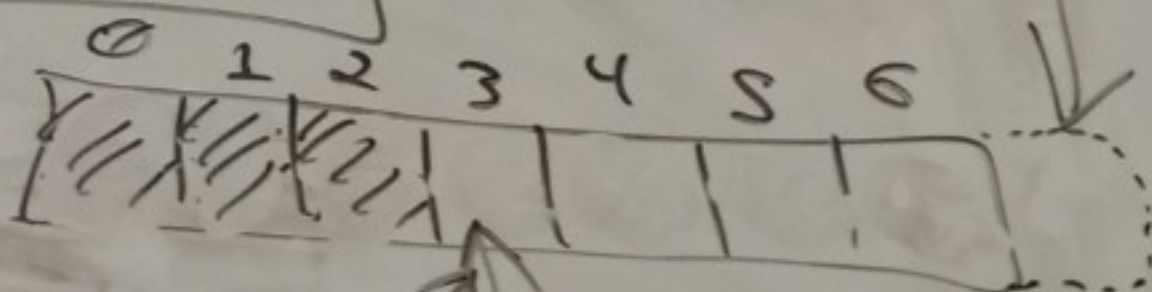
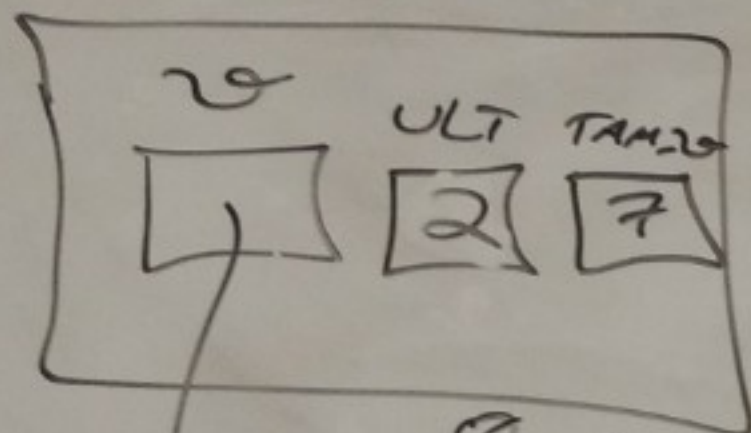
} // CONJUNTO



40 *



C



início

int *p;

p = &(v[0]);

... Povoar C...

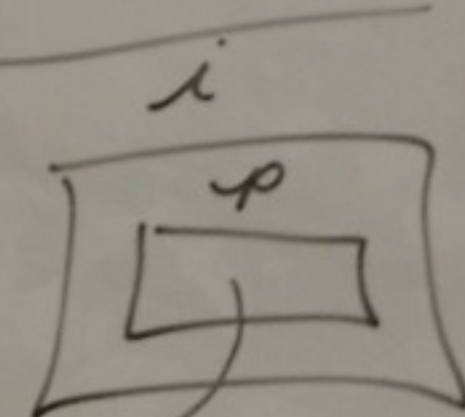
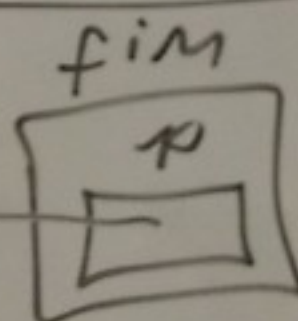
auto fim = C.fim();

for(auto i = C.inicio(); i != fim; ++i)

cout << ' ' << *i;

C.terminar();
} //main

main



2. INTRO

um cont

conjunto

é difer

apenas um

elemento

retorna

ações com

DICIONÁRIOS

2. INTRODUÇÃO: UM DICIONÁRIO É UM CONTÊINER, ASSIM COMO UM CONJUNTO, MAS A OPERAÇÃO DE BUSCA É DIFERENTE DA OPERAÇÃO "PERTENCE": AO INVÉS DE RECEBER UM ELEMENTO INTEIRO, RECEBE APENAS UMA **CHAVE**, QUE IDENTIFICA O ELEMENTO DE FORMA ÚNICA, E ENTÃO RETORNA OS ^(VALOR) **DADOS-SATÉLITE** ("INFORMAÇÕES COMPLEMENTARES") CORRESPONDENTES.

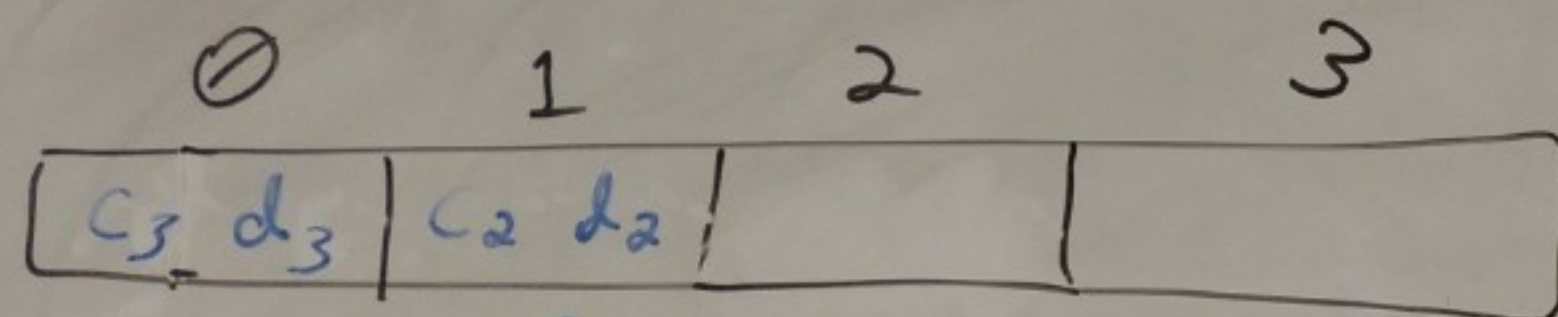
AS PRINCIPAIS OPERAÇÕES DE UM DICIONÁRIO SÃO:

a) $\text{INSERIR}(D, c, d)$: INSERE A CHAVE c E O DADO-SATÉLITE d COMO UM NOVO PAR NO DICIONÁRIO D .

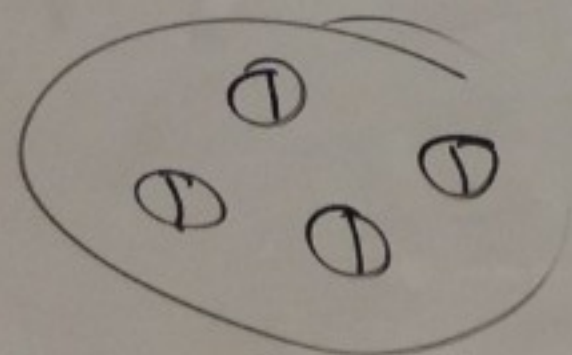
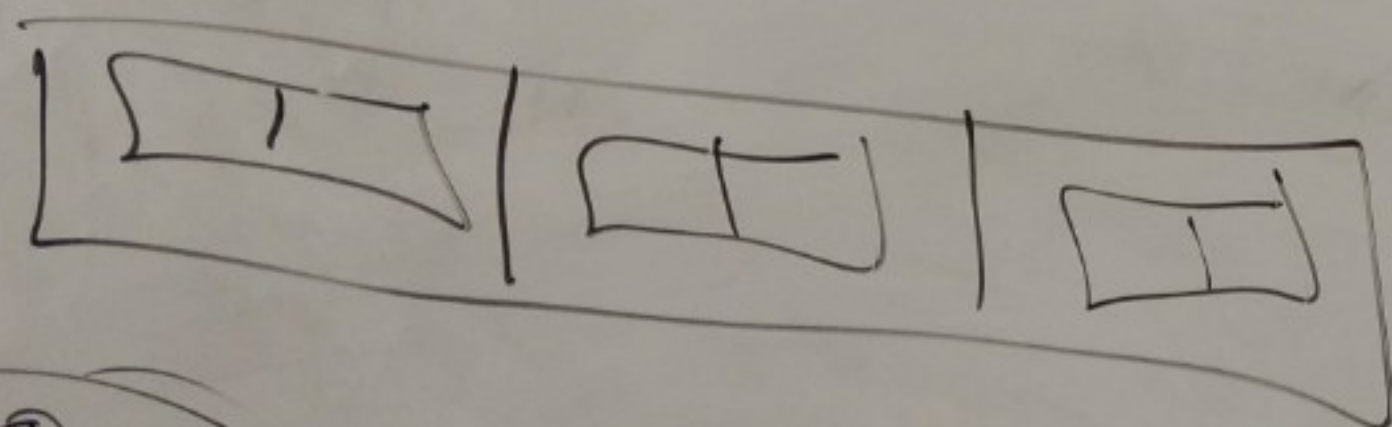
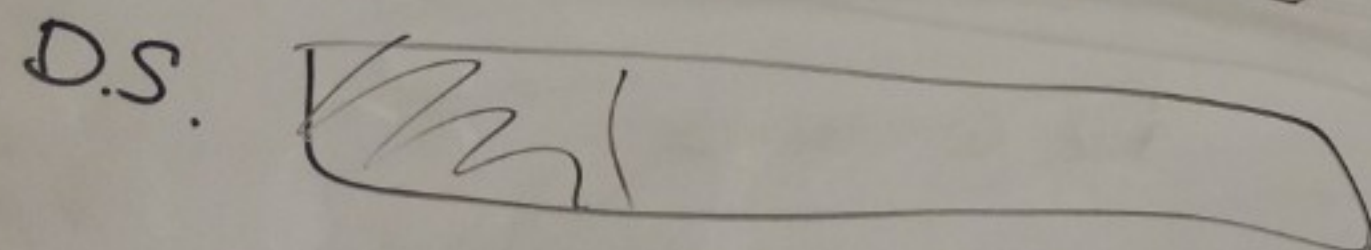
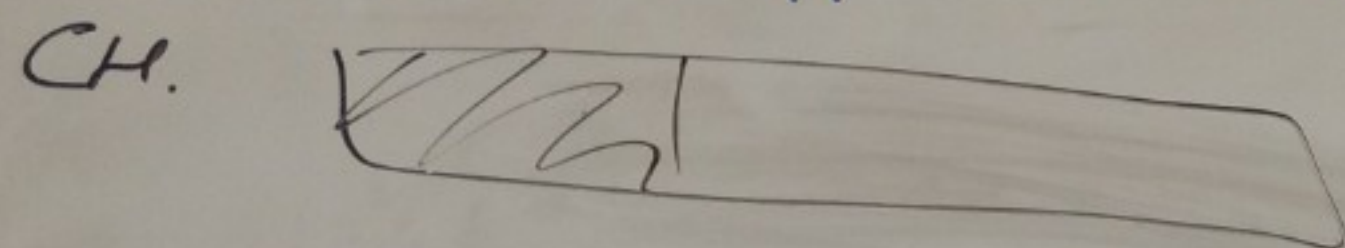
b) $\text{PROCURAR}(D, c)$: RETORNA O DADO SATÉLITE CORRESPONDENTE À CHAVE c NO DICIONÁRIO D .

c) $\text{REMOVER}(D, c)$: REMOVE DE D O PAR/ELEMENTO DE CHAVE c .

3. REPRESENTAÇÃO:



$C\langle T \rangle \parallel D\langle TC, TD \rangle$



4. Uma implementação ("Dicionario.hpp")

// GUARDAS DE INCLUSÃO...

#INCLUDE <NEW>

USING STD::NOTHROW;

TEMPLATE < typename X, typename Y >

STRUCT PAR { X x; Y y; };

TEMPLATE < typename TC, typename TD >

STRUCT Dicionario

{

PAR < TC, TD > *v;

INT ULT, TAM_v;

// Dicionário

BOOL INICIALIZAR ()

```
{  
    v = NEW(NOTHROW)  
    PAR < TC, TD > [1];  
    if (v == NULLPTR)  
    { RETURN TRUE; }  
    ULT = -1; TAM_v = 1;  
    RETURN FALSE;  
}
```

VOID TERMINAR ()

```
{  
    DELETE[] v;  
}
```

BOOL VAZIO ()

```
{  
    RETURN (ULT == -1);  
}
```

// Dicionário

PAR < BOOL, TD > PROCURAR (TC e)

```
{  
    FOR (int i = 0; i <= ULT; ++i)  
    {  
        if (v[i].x == e)  
        {  
            PAR < BOOL, TD > p;  
            p.x = TRUE;  
            p.y = v[i].y;  
            RETURN p;  
        }  
    } // FOR
```

PAR < BOOL, TD > p;

p.x = FALSE;

RETURN p;

```
}
```


// Dicionário

PAR<BOOL, BOOL> INSERIR(τ C, τ D d)

{
 ↳ ERRO DE ALOCAÇÃO?
 ↳ JÁ PERTENCE?

PAR<BOOL, τ D> p = PROCURAR(c);

// AUTO p = ...

if (p.x)

{

PAR<BOOL, BOOL> π ;

$\pi.x = \text{TRUE}$; // " $c \in D$ "

RETURN π ;

}

if (ULT == TAM - τ - 1)

{

if (REDIMENSIONAR($2 * \text{TAM} - \tau$))

{ PAR<BOOL, BOOL> π ;

$\pi.x = \text{FALSE}$;

$\pi.y = \text{TRUE}$; RETURN π ;

}

}

// Dicionário, INSERIR

++ULT;

τ [ULT].x = c;

τ [ULT].y = d;

PAR<BOOL, BOOL> $\pi = \{\text{FALSE}, \text{FALSE}\}$;

RETURN π ;

} // INSERIR

...