GRR20197152 Guilherme Costa Pateiro
GRR20190485 Gustavo Henrique da Silva Barbosa

Definições gerais

- **Constantes:**

```c
#define TAM 11
#define OPERACAO_REALIZADA 1
#define ERRO 2
```

TAM: Tamanho das listas.
OPERACAO_REALIZADA: Resposta afirmativa na saída das funções, muito utilizada para verificar se houve ou não uma inserção ou remoção.
ERRO: resposta negativa na saída das funções.

- **Struct (TAD Hash):**

```c
typedef struct {
    int elemento;
    int excluido;
} hash;

typedef struct {
    hash t1[TAM];
    hash t2[TAM];
} cockoo_h;
```

A struct utilizada nas funcoes foi a “cockoo_h”, formada por 2 arrays da struct “hash” de tamanho TAM
A struct “hash” é formada por 2 campos, o campo “elemento”, que contém o item que irá ser guardado no endereço da tabela, e um campo "excluido" que serve para verificar se o elemento desse endereço está acessível, caso o endereço não tenha sido preenchido com um valor ou caso esse elemento seja removido da tabela, o valor de “excluido” será 1 .

Insere:

```c
int insere (int valor, cockoo_h *tabela){
    int resposta;
    resposta = insere_tabela(valor, tabela->t1, h1);
    if(resposta == ERRO){
        hash item = (tabela->t1)[h1(valor)];
        insere_tabela(item.elemento, tabela->t2, h2);
        exclui_tabela(item.elemento, tabela->t1, h1);
        insere_tabela(valor, tabela->t1, h1);
    }
    return OPERACAO_REALIZADA;
}
```

O algoritmo de inserção, segue basicamente o fluxo de: tentar inserir em “t1”, caso ocorra erro, o elemento de “t1” é passado para “t2”, e o valor é novamente inserido em “t1”.

A maior particularidade deste código é o fato que utilizamos as funções auxiliares “insere_tabela” e “exclui_tabela”. A função “insere_tabela” recebe como parâmetros o valor a ser inserido, a tabela em que esse valor será inserido e a função hash. Esta função, a partir dos valores passados, tenta inserir na tabela hash, porém, se houver algum valor no endereço calculado, ou seja, se ele não foi "excluído" e houver um valor válido nele é retornado “ERRO”, caso contrário, o valor é inserido.

Exclusão

```
int exclui (int valor, cockoo_h *tabela){
    int resposta;
    resposta = exclui_tabela(valor, tabela->t1, h1);
    if(resposta == ERRO){
        exclui_tabela(valor, tabela->t2, h2);
    }
    return OPERACAO_REALIZADA;
}
```

A função de exclusão, possui um funcionamento similar a de inclusão, ou seja, primeiramente realizada a tentativa de excluir o item de “t1”, porém se isso não for possível, é realizada a tentativa de excluir o item de “t2”.

Como função auxiliar, temos “exclui_tabela”, sendo que esta função recebe como parâmetros, o valor a ser excluído, a tabela e a função de hash. Esta verifica se o valor do endereço é igual ao presente no endereço, se sim a operação é realizada, caso contrário é retornado erro.

Imprime:

```
int imprime_tabelas(cockoo_h *tabela){
    declaracao de variaveis;
    for (j = 0; j < TAM*2; j++){
        encontra_menor_valor(T1,T2);
        if (elemento de T1 <= elemento de T2){
            imprime(item T1);
        }
        else if (elemento de T1 > elemento de T2){
            imprime(item T2);
        else
            return ERRO;
    }
    return OPERACAO_REALIZADA;
```

A impressão funciona percorrendo ambas as listas e encontrando o menor elemento não excluído delas, então é feito a comparação entre os 2 elementos encontrados de cada lista, o menor entre os 2 elementos é impresso na saída e o elemento é marcado como excluído da tabela, para evitar que este interfira com a procura de novos valores no reinicio do “for”. Uma característica importante dessa função é que ela exclui a lista inteira (no caso marca todos os elementos da lista com excluído).