# Relatório Algoritmos III - HASH

Nícolas André Rizzardi - GRR20206152

Pedro Ernesto Viana Heberle - GRR20205519

# Definições

#define COL 2

#define M 11

# Estruturas

A estrutura é unicamente utilizada para poder fazer a impressão de maneira ordenada, tem um vetor com as chaves existentes na hash e um inteiro para armazenar seu próprio tamanho.

typedef struct array{

int entradas[COL\*M];

int tam;

} array\_t;

Para as tabelas hash, utilizamos uma única matriz de 2 colunas por M=11 linhas para representar as duas tabelas diferentes, onde cada coluna é uma tabela com função hash específica. Para posições vazias o valor da célula é -1, já para posições excluídas, o valor da célula é -2.

# Funções Principais

**Função utilizada para criar a estrutura que armazena os valores da Hash.**

**Retorna a struct inicializada.**

array\_t \*new\_arrayt();

**Função void para inserir um valor da hash na struct de entradas**.

void insere\_valor(int valor, array\_t \*vetor);

**Função void para remover um valor da hash na struct de entradas.**

void remove\_valor(int valor, array\_t \*vetor);

**Função void que recebe a matriz de hash e inicializa com o valor padrão que definimos como posição vazia.**

void init\_table (int ht[COL][M]);

**Função que insere uma chave nas tabelas hash, dentro dessa função ocorre a verificação de função hash da primeira tabela e caso a posição esteja ocupada envia o valor antigo para a segunda tabela, liberando espaço para a nova chave na primeira tabela**

void include\_key (int key, int ht[COL][M]);

**Função que busca a chave e caso a encontre remove a mesma da tabela hash.**

**A função tenta remover da segunda tabela antes de partir para a primeira, mas caso a chave esteja na primeira ele substitui o valor da chave pelo valor que indica um rastro de que a posição já possuiu um valor. Retorna um inteiro para confirmar se o valor foi removido com sucesso ou não**

int remove\_key (int key, int ht[COL][M]);

**Função void utilizada para imprimir as chaves presentes na tabela hash, ordenadas de maneira crescente e apontando qual tabela e qual a posição se encontra.**

void print\_table(array\_t \*vetor, int ht[COL][M]);

# Funções Internas

**Função que define a posição na tabela 1 em que se insere a chave.**

int hash\_t1 (int key);

**Função que define a posição na tabela 2 em que se insere a chave.**

int hash\_t2 (int key);

**Função que retorna um inteiro que corresponde à tabela em que a chave se encontra (nesse caso, 1 ou 2).**

int find\_key (int key, int ht[COL][M]);