

Tópico: **Hashing**

1. Implemente as funções para fazer inserção, busca e remoção na tabela *Hash* conforme os protótipos definidos abaixo, considerando:

- (a) o número de matrícula seja utilizado como chave
- (b) o tratamento de colisão é simplesmente imprimir uma mensagem “posição ocupada” e ignorar a inserção

```
#define TAM 1000 // tamanho da tabela

typedef int TipoChave;

typedef struct {
    int matricula;
    char nome[50];
} TipoRegistro;

typedef struct {
    TipoChave k; // chave
    TipoRegistro r; // registro armazenado
    int ocupado; // indica se o slot está ocupado
} slot;

typedef slot TabelaHash[TAM];

void inicializaTabelaHash(TabelaHash T);

void inserir(TabelaHash T, TipoChave k, TipoRegistro r);

//retorna a posição do registro de chave k na tabela hash T
// ou -1 caso a chave não esteja presente.
int buscar(TabelaHash T, TipoChave k);

void remover(TabelaHash T, TipoChave k);

int hash(int k){
    return k%TAM;
}
```

2. Altere o código do exercício anterior para que seja feito o tratamento de colisões por endereçamento aberto usando:
  - (a) sondagem linear
  - (b) sondagem quadrática
  - (c) *hash* duplo
3. Altere o código do exercício 1 para que seja feito o tratamento de colisões por encadeamento.
4. Considere o método da multiplicação usado na definição da função *hash*  $h(k) = \lfloor m(kA \bmod 1) \rfloor$ , onde  $A = (\sqrt{5} - 1)/2$  e  $m = 1000$ . Calcule os valores de  $h$  para as chaves 31, 32, 33, 34 e 35.

5. Considere o método da divisão para definição de funções *hash*:  $h(k) = k \bmod m$ . Encontre 3 chaves diferentes que tenham o mesmo valor *hash*, considerando:

- (a)  $m = 143$
- (b)  $m = 157$

Encontre uma fórmula geral para calcular tantas chaves quanto se queira cujo valor *hash* seja igual ao valor *hash* de uma dada chave  $k$ .

6. Suponha uma tabela *hash* de tamanho  $M=100$  para armazenar chaves no intervalo  $[1 - 9999]$ . Insira as seguintes chaves nessa tabela: 3579, 7279, 8879, 1179, 1233, 8733, 111, 3211, 5711, 2100, 5600, 382 nessa ordem, considerando diferentes métodos de resolução de colisões:

- (a) encadeamento usando listas ligadas
- (b) endereçamento aberto:
  - i. sondagem linear:  $h(k, i) = (h'(k) + i) \bmod M$  e  $h'(k) = k \bmod M$
  - ii. sondagem quadrática:  $h(k, i) = (h'(k) + i^2) \bmod M$  e  $h'(k) = k \bmod M$
  - iii. sondagem quadrática:  $h(k, i) = (h'(k) + 2i + 5i^2) \bmod M$  e  $h'(k) = k \bmod M$
  - iv. *hash* duplo:  $h(k, i) = h_1(k) + ih_2(k)$ ,  $h_1(k) = k \bmod M$  e  $h_2(k) = 7 - (k \bmod 7)$

7. Segundo dados do IBGE (ano base: 2016), o Brasil tem 5.570 municípios. Considere o problema de armazenar informações relacionadas aos municípios brasileiros, onde o nome do município será usado como chave. Proponha uma função *hash* que utilize alguma combinação dos códigos ASCII das letras do nome do município e estado tal que cidades homônimas e também anagramas sejam mapeados para valores diferentes na função *hash*.