

# Este trabalho é sobre tabelas de dispersão (ou tabelas hash)

com tratamento de colisão chamado de “linear probing” (também conhecido como sondagem linear).

## Os parâmetros importantes são:

- $N$  é o número de posições na tabela de hash (tamanho da tabela)
- $M$  é uma variável que contém a quantidade de chaves atualmente incluídas na tabela
- $\text{Alfa}$  é o fator de carga (*load factor*) que é dado por  $M/N$

Como a cada inserção o valor da variável  $M$  é acrescentada, o valor de  $\text{Alfa}$  também deve ser recalculado.

- Considere que a função hash é  $h(k) = k \% N$  (onde  $k$  é a chave)

No caso da técnica de *linear probing*, o valor do índice correspondente a chave que sofreu colisão é incrementado até que uma posição disponível seja encontrada, isto é,  $h'(k) = h(k) + i$  onde  $i$  pode ter o valor 1, 2, 3, e assim sucessivamente.

Considere que a Tabela Hash contém somente os campos **matrícula** (chave) e **primeiro nome** (uma palavra somente).

Vamos assumir que a Tabela Hash deve ser mantida sempre com um fator de carga abaixo de um limite  $W$ , ou seja, após cada inserção, deve ser verificado se o fator de carga ( $\text{Alfa}$ ) é igual ou superior a  $W$  e que, neste caso, deve ser feito o redimensionamento da tabela

Assuma que o redimensionamento deve **dobrar o valor de  $N$**  e que, portanto, a Tabela Hash deve ser refeita ou seja, *todas as inserções já feitas anteriormente devem ser refeitas*. Tais re-inserções serão feitas na ordem dos índices da Tabela Hash anterior.

# Entrada

---

A entrada de dados será feita em três etapas.

## etapa 1

---

consiste na entrada de três valores inteiros positivos.

- O primeiro é o *tamanho inicial da tabela* (  $N$  ), onde  $1 \leq N \leq 500$
- O segundo é a *quantidade de chaves* (  $M$  ) a serem entradas no total (pode ser maior que  $N$  )
- O terceiro é o *percentual do fator de carga* (  $W$  ), que é o limite do fator de carga para que haja o redimensionamento  
se  $W = 63$  implica que, quando  $\text{Alfa} \geq 0.63$  o redimensionamento deve ser feito

## etapa 2

---

consiste em fornecer as  $M$  chaves que deverão ser **incluídas** na Tabela Hash e, caso necessário, deve ser usado o tratamento de colisões *linear probing*

Também deve ser observado a necessidade de redimensionamento

## etapa 3

---

consiste em fornecer  $K$  chaves que serão **buscadas** na Tabela Hash.

# Saída

---

A saída será feita em duas etapas.  
Ao término da leitura de todas as chaves,

## etapa 1

---

- A impressão do valor atual de  $N$  da Tabela Hash
- Em seguida, deve ser impresso o **índice**, **chave** e **nome** separados por espaço.  
Caso o elemento não tenha um valor, deve ser impresso `\` (*contrabarra*)  
Considere todos os redimensionamentos já feitos

## etapa 2

---

Após a finalização da etapa anterior

- deverá ser impresso `SIM` para cada chave encontrada na Tabela Hash, e `NAO` caso contrário.

# Exemplo

## Entrada

```
10 11 60
1234 Erick
3234 Larissa
1324 Fabio
3333 Maria
3019 Luciana
2786 Juliana
4545 Pedro
1111 Felipe
2222 Joao
3334 Paulo
3312 Jose
5
3333
3222
3019
4120
1234
```

## Saída esperada

```
20
0 \
1 \
2 2222 Joao
3 \
4 1324 Fabio
5 4545 Pedro
6 2786 Juliana
7 \
8 \
9 \
10 \
11 1111 Felipe
12 3312 Jose
13 3333 Maria
14 1234 Erick
15 3234 Larissa
16 3334 Paulo
17 \
18 \
19 3019 Luciana

SIM
NAO
SIM
NAO
SIM
```