

Hashing com Sondagem Linear

Sua tarefa neste exercício é implementar *hashing* com a técnica de resolução de colisões chamada *sondagem linear*.

Para tanto, considere a seguinte representação de tabela *hash*:

```
typedef struct {  
    int *tb; // tabela hash  
    int M;   // tamanho da tabela hash  
    int N;   // total de chaves presentes na tabela  
} TH;
```

em que `table` é o vetor que representa a tabela *hash*, `M` é o tamanho da tabela e `N` é o total de chaves presentes na tabela.

Você deve submeter um arquivo contendo três funções:

1. A função

```
void THinsere (TH *h, int ch);
```

deve inserir a chave `ch` na tabela *hash*. Nesta função de inserção:

- você deve tratar as colisões com **sondagem linear**, de tal forma que a sondagem procure a próxima posição disponível à direita da posição *hash* da chave, com saltos de uma casa.
- caso a tabela *hash* tenha taxa de ocupação de mais de 50% (ou seja, se $N > M/2$), antes de inserir um novo elemento você deve redimensionar a tabela. Para tanto, você deve consultar o próximo primo usando uma função já implementada `int aumentaTamanho (int M)` que recebe o tamanho atual da tabela *hash* e retorna o novo tamanho. **Esta função deve ser declarada no preâmbulo do seu código.**
- *cuidado para não ter chaves duplicadas.*

2. A função

```
int THremove (TH *h, int ch);
```

deve remover a chave `ch` da tabela *hash*. Esta função deve retornar zero, se a chave foi encontrada e removida, ou `-1` caso a chave não estivesse presente na tabela. Além disso, se a chave for encontrada e removida, deve-se reinserir na tabela *hash* as chaves que forem necessário.

3. A função

```
int THbusca (TH *h, int ch);
```

deve buscar ocorrência da chave `ch` na tabela *hash*. Ela deve retornar 1 se a chave for encontrada, ou 0, caso contrário.

Restrições

- Considere que o **universo de chaves é composto por inteiros não negativos**.
- Use como símbolo de vazio o número `-1`.
- Você deve utilizar a *função hash modular* na sua implementação.
- Você deve submeter um código que contenha **apenas as três funções especificadas mais alguma função auxiliar que você julgue necessária à sua solução**. Você não deve submeter a função `main`.

Author: John L. Gardenghi