

AED2 2023 (1s) - EXERCÍCIO 9 - TABELA HASH

Instruções:

1. E/S: tanto a entrada quanto a saída de dados devem ser "secas", ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas (veja o exemplo abaixo);
2. Identificadores de variáveis: escolha nomes apropriados;
3. Documentação: inclua cabeçalho, comentários e indentação no programa;
4. Submeta o programa no sistema judge:
<https://judge.unifesp.br/aediiS1A23/>.

Descrição: Implemente as operações básicas (1) pesquisa, (2) inserção e (3) remoção de chaves em uma tabela *Hash* com endereçamento aberto. Escreva procedimentos separados para cada operação. Utilize o duplo mapeamento, visto nos slides da aula, como seu método de sondagem e tratamento de colisões. Para isso, lembrando que a função de transformação é dada por:

$$h(k, i) = (h_1(k) + ih_2(k)) \bmod m$$

e sabendo que o tamanho m da tabela é um número primo, utilize as seguintes funções auxiliares:

$$h_1(k) = k \bmod m$$

$$h_2(k) = 1 + (k \bmod (m - 1))$$

Considere as seguintes condições:

1. A complexidade de cada procedimento implementado deve ser a mesma apresentada em aula;
2. O código-fonte **deve** ser escrito em C/C++ ou Java;
3. **Toda** memória alocada dinamicamente (C/C++) deve ser desalocada;

Solução que violem essas condições **não** serão aceitas.

ENTRADA:

A entrada consiste de 3 linhas, cada qual contendo números inteiros:

1. A primeira linha contém um único número inteiro primo (m) indicando o tamanho da tabela de *Hash* a ser utilizada;
2. A segunda linha contém uma lista de números inteiros não-negativos, separados por espaços, que devem ser inseridos na tabela *Hash*. O final dessa lista é marcado pela presença do valor -1, que não deve ser inserido;

3. A terceira linha contém um número inteiro a ser utilizado como chave de pesquisa na tabela. No caso de uma pesquisa com sucesso, ou seja, o valor está na tabela, **todas as ocorrências** desse valor deverão ser removidas da tabela. Caso contrário, ou seja, o valor não está na tabela, apresente o texto "*Valor nao encontrado*" (texto deve ser escrito exatamente como mostrado, sem acentuação). A cada item removido, utilize um código especial, diferente dos dados de entrada, para marcar a posição adequadamente, pois poderá ser necessário recuperar essa informação na saída;

SAÍDA:

A saída pode conter uma ou duas linhas:

- Ela terá duas linhas quando o valor pesquisado não estiver na tabela, pois a informação "*Valor nao encontrado*" deve ser apresentada na primeira linha da saída;
- A segunda linha – ou primeira, no caso de sucesso na pesquisa – deve apresentar o conteúdo da tabela *Hash* **após** a operação de busca/remoção. As chaves devem ser separadas por espaço. Posições não utilizadas devem ser representadas pelo caractere '\', e posições onde um elemento foi removido, se existirem, pelo caractere 'D'. Veja os exemplos fornecidos.

Exemplos de entrada e saída:

- *input01:*

Entrada	Saída
7 3 4 9 8 1 2 -1 1	2 8 9 3 4 D \

Tabela 1: Exemplos de entrada e saída 01

- *input02:*

Entrada	Saída
11 11 22 14 21 -1 12	Valor nao encontrado 11 \ \ 22 \ \ \ \ 14 \ 21

Tabela 2: Exemplos de entrada e saída 02

- *input03:*

Entrada	Saída
5 3 2 1 4 4 -1 4	\ D 1 2 3 D

Tabela 3: Exemplos de entrada e saída 03