

IMD0029 - Estrutura de Dados Básicas 1 –2023.1 – Prova 03

Prof. Eiji Adachi M. Barbosa

Nome: _____

Matrícula: _____

ANTES DE COMEÇAR A PROVA, leia atentamente as seguintes instruções:

- Esta é uma prova escrita de caráter individual e sem consultas a pessoas ou material (impresso ou eletrônico).
- A prova vale 5,0 pontos e o valor de cada questão é informado no seu enunciado.
- **Preze por respostas legíveis, bem organizadas e sucintas.**
- As respostas devem ser fornecidas preferencialmente em caneta. Respostas fornecidas a lápis serão aceitas, mas eventuais questionamentos sobre a correção não serão aceitos.
- Celulares e outros dispositivos eletrônicos devem permanecer desligados durante toda a prova.
- Desvios éticos ou de honestidade levarão à anulação da prova do candidato (nota igual a zero).

Questão 1 (1,5 ponto): O TAD Multi-Conjunto, assim como o TAD Conjunto, não admite a repetição de elementos. No entanto, um Multi-Conjunto mantém uma contagem de quantas vezes um determinado elemento foi inserido; esta contagem é chamada de ‘multiplicidade’. Por exemplo: o Multi-Conjunto $A = \{X, X, Y\}$ possui os elementos X e Y, mas o elemento X possui multiplicidade igual a 2 e o elemento Y possui multiplicidade igual a 1. Já o Multi-Conjunto $B = \{X, X, Y, Y, X\}$ também possui os elementos X e Y, mas o elemento X possui multiplicidade igual a 3 e o elemento Y possui multiplicidade igual a 2. Neste exemplo, apesar de possuírem os mesmos elementos X e Y, os Multi-Conjuntos A e B não são iguais, pois a multiplicidade de seus elementos não é igual.

Nesta questão, assuma que você possui à sua disposição um TAD Dicionário já implementado corretamente e que provê as seguintes operações:

```
Dicionario::Dicionario ( ) // Construtor
bool Dicionario::Inserir( C chave, V valor )
bool Dicionario::Remover( C chave)
bool Dicionario::Buscar( C chave ) // Retorna o valor associado a chave, ou null caso contrário
```

Com base neste TAD Dicionário, implemente uma função que recebe como entrada um array de strings e retorna um dicionário que representa um multi-conjunto contendo cada string única do array e suas respectivas multiplicidades. Por exemplo, se a entrada for igual ao seguinte array de strings $S = \{ "X", "X", "Y", "X", "Y", "Z" \}$, a saída deverá ser um dicionário igual a $D = \{ ("X", 3), ("Y", 2), ("Z", 1) \}$.

Sua solução deverá ter a seguinte assinatura:

```
Dicionario construirMultiConjunto( int array[], int tamanhoArray );
```

Questão 2 (1,5 ponto): No contexto de Tabela Hash, explique:

- a) O que é uma colisão? (0,5 ponto)
- b) De que forma a estratégia do encadeamento externo resolve a colisão? (1,0 ponto)

Questão 3: (2,0 pontos) Para cada uma das afirmações a seguir, marque V (verdadeiro) ou F (falso), **justificando sucintamente** todas as suas respostas. Marcações de V ou F **sem justificativas não serão aceitas.**

1 – () Em uma Tabela Hash implementada usando a estratégia de endereçamento aberto com sondagem linear sem redimensionamento dinâmico, existe um limite máximo para a quantidade de elementos que podem ser armazenados na estrutura.

2 – () Em uma Tabela Hash implementada usando a estratégia de encadeamento externo sem redimensionamento dinâmico, existe um limite máximo para a quantidade de elementos que podem ser armazenados na estrutura.

3 – () Em uma Tabela Hash implementada usando a estratégia de endereçamento aberto com sondagem linear, o método de redimensionamento da tabela tem complexidade assintótica linear no pior caso e complexidade assintótica constante no melhor caso.

4 – () O pior caso da busca em uma Tabela Hash implementada usando a estratégia de endereçamento aberto com sondagem linear é mais eficiente, do ponto de vista da complexidade assintótica, do que o pior caso da busca em uma Tabela Hash implementada usando a estratégia de encadeamento externo.