

IMD0029 - Estrutura de Dados Básicas 1 –2019.2 – 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 mesmo 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:

```
Dictionary::Dictionary ( ) // Construtor
bool Dictionary::Put( K key, V value ) // Inserir
bool Dictionary::Remove( K key ) // Remover
bool Dictionary::Get( K key ) // Retorna o valor associado a key, 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 a $S = \{X, X, Y, X, Y, Z\}$, a saída deverá ser um dicionário igual a $D = \{ (X, 3), (Y, 2), (Z, 1) \}$.

Sua solução deverá ter a seguinte assinatura:

```
Dictionary buildMultiSet( int array[], int arraySize );
```

Questão 2 (1,5 ponto): Em sala de aula, vimos que a função hash (ou função de dispersão) é parte central na implementação da estrutura Tabela de Dispersão. Explique para que serve a função hash no contexto de uma Tabela de Dispersão. Explique também o que é colisão e como pode ser tratada.

Questão 3 (1,0 ponto): Em sala de aula, vimos como implementar a funcionalidade de inserir elementos numa Tabela de Dispersão seguindo duas estratégias: (i) *lazy* e (ii) *look-ahead*. Explique como funciona cada uma dessas estratégias.

Questão 4 (1,0 ponto): Em sala de aula, vimos o conceito de Tabela de Dispersão com Redimensionamento Dinâmico (também chamada de Tabela de Dispersão Elástica). A respeito desta tabela, explique qual problema ela tenta solucionar e como ela funciona.