

IMD0029 - Estrutura de Dados Básicas 1 – Turma 04 – 2016.2 – 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 10,0 pontos e o valor de cada questão é informado no seu enunciado.
- Preze por respostas legíveis, bem organizadas e simples. Não é necessário seguir a notação de C++, embora seja permitido usar tal notação. Pode usar uma notação mais simples, estilo pseudo-código.
- Leia atentamente às observações em cada questão. Elas fazem parte do enunciado.
- 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: (3,0 pts) Considere uma Tabela de Dispersão definida conforme as estruturas a seguir:

<pre>HashTable { HashEntry data[]; int quantity; int size; }</pre>	<pre>HashEntry { K key; // chave de tipo K V value; // valor de tipo V }</pre>
--	--

Considere ainda que esta Tabela de Dispersão resolve suas colisões por meio do método do *Endereçamento Aberto com Sondagem Linear*. Neste contexto, implemente para esta Tabela de Dispersão uma função de redimensionamento dinâmico, a qual poderá ser usada tanto para aumentar ou diminuir o seu tamanho. A sua função deverá ter a seguinte assinatura:

```
void HashTable::resize(int newSize)
```

Questão 2: (2,0 pts) A Tabela de Dispersão é uma estrutura de dados bastante utilizada em soluções que não requerem sequência entre os dados ordenados e que requerem consultas eficientes aos dados armazenados. Neste contexto, implemente uma função de busca para uma Tabela de Dispersão definida seguindo as estruturas definidas na Questão 1. Nesta questão, considere que se retorna nulo quando a busca tem insucesso e que a tabela resolve suas colisões pelo método do *Endereçamento Aberto com Sondagem Quadrática*. A sua função deverá ter a seguinte assinatura:

```
// Recebe uma chave (key) e retorna o seu valor V correspondente
V HashTable::get(K key)
```

Obs.1: Nesta questão, você poderá empregar qualquer estratégia de endereçamento aberto com sondagem quadrática, desde que seja de fato quadrática. Ou seja, não precisa se preocupar de que forma as chaves foram inseridas anteriormente na tabela em que serão feitas as buscas.

Questão 3: (3,0 pts) Um anagrama é um jogo de palavras em que tentamos criar outras palavras por meio do rearranjo das letras de uma determinada palavra base. As novas palavras criadas devem empregar todas as letras originais da palavra base exatamente uma vez. Um exemplo bastante conhecido de anagrama é o nome da

personagem Iracema, do escritor José de Alencar, que é um anagrama da palavra America. Nesta questão, você implementará uma solução que irá auxiliar uma pessoa a encontrar anagramas de uma determinada palavra. Tal solução terá como base um dicionário de anagramas. Para construir este dicionário, implemente uma função que irá receber como entrada um *array* de palavras que servirão como base para a sua solução “aprender” possíveis anagramas e construir o dicionário desejado. Esta função deverá ter a seguinte assinatura:

```
Dictionary learn(string words[], int arraySize)
```

Faz parte da solução desta questão a definição da estrutura `Dictionary`. Portanto, além de implementar a função `learn`, você também deverá mostrar sua definição da estrutura `Dictionary`. Não é necessário implementar as operações básicas sobre o dicionário (Inserir, Remover e Buscar); pode considerar que estas operações já estão implementadas e podem ser reusadas. O que é requerido nesta questão é a definição da estrutura interna do dicionário. Ou seja, quais seus atributos e tipos de cada atributo.

Obs.1: Nesta questão, você poderá empregar funções auxiliares de alto nível, desde que fique claro qual o propósito destas funções, seja por meio de comentários, ou pelo próprio nome da função.

Questão 4: (2,0 pts) Considere que numa determinada solução fez-se necessário definir uma Tabela de Dispersão cuja chave é do tipo `Pair`, conforme a seguinte estrutura:

```
Pair
{
    string first;
    string second;
}
```

A estrutura `Pair` nada mais é do que uma estrutura que encapsula duas strings. Nesta questão, você deverá implementar uma função de dispersão (função *hash*) que opere sobre a estrutura `Pair` definida acima. Sua função deverá ter a seguinte assinatura:

```
long HashTable::hash(Pair p)
```

Obs.1: Nesta questão, você deverá de fato implementar a função de dispersão em seus detalhes. Considere que não há outras funções de dispersão auxiliares que possam ser reusadas.