

Curso <i>Bacharelado em Engenharia de Software</i>			Campus <i>Samambaia – Campus II</i>
Disciplina <i>AOB – Algoritmos: Ordenação e Busca</i>			
Nome do(a) acadêmico(a)			Assinatura
Nº de matrícula	Turma <i>3º período</i>	Data de Entrega <i>26/05/2016</i>	Professor(a) <i>Wanderley de Souza Alencar</i>

I. Dos Objetivos

Esta *avaliação formativa* (AF) está associada do *primeiro bimestre* do semestre letivo 2016/1 e tem três objetivos:

1. Tendo em vista de que a classe de algoritmos abordados até o momento na disciplina *Algoritmos: Ordenação e Busca* (ou seja, os algoritmos de *pesquisa* ou *busca* em memória principal) é de notória importância para a Informática e Computação, é necessário proporcionar ao acadêmico da disciplina a possibilidade de aprofundar seu conhecimento a respeito deles quando aplicados sobre as estruturas de dados conhecidas pelo nome de *tabelas*;
2. Permitir que, por meio da implementação, e consequente processo de experimentação dos supramencionados algoritmos, os acadêmicos percebam – de forma prática – o efeito do uso de algoritmos com diferentes *complexidades de tempo* para a realização de uma mesma operação sobre uma estrutura de dados definida, neste caso, sobre *tabelas*. A partir desta percepção, tornem-se críticos quanto a escolha de algoritmos a serem por eles utilizados, seja no âmbito acadêmico ou profissional;
3. Aprimorar o domínio quanto ao uso de uma linguagem e/ou ambiente de programação para realizar a implementação de algoritmos.

II. Das Equipes de Acadêmicos e da Atribuição de Nota

Para realização desta AF, os acadêmicos se aglutinarão em *equipes* que poderão ter de 1 (um) a 5 (cinco) acadêmicos, conforme livre organização definida entre os partícipes da turma e registradas na área da disciplina na Plataforma de Educação a Distância do INF/UFG, localizada no endereço eletrônico <https://ead.inf.ufg.br>.

A *nota* desta AF será atribuída à equipe e, portanto, todos os integrantes obterão notas idênticas. Caberá à própria equipe gerir a distribuição das atividades a serem realizadas por seus integrantes para a consecução da AF e sua entrega conforme as especificações a seguir apresentadas.

II. Da Especificação da Avaliação

Esta AF consiste na *implementação e experimentação* de algoritmos para a realização das operações básicas sobre *tabelas*, ou seja, das operações de (1) inserção; (2) remoção; e (3) pesquisa ou busca. Também serão exigidos algoritmos complementares, ou seja, de suporte à implementação dos anteriormente mencionados como, por exemplo, um algoritmo para a apresentação de resultados, na tela, para o usuário.

Deverá haver *duas diferentes implementações* para a estrutura de dados denominada de *tabela*:

1. Como um *lista linear encadeada* (ou *linear linked list*) cujas *chaves primárias* são números naturais, sendo que o tamanho da tabela deverá variar de 10^1 a 10^6 elementos: 10^1 , 10^2 , 10^3 , 10^4 , 10^5 e 10^6 .

Cada *nó* (ou *nódulo*) da lista linear encadeada deverá ter a seguinte estrutura:

<i>chave primária</i>	<i>dado</i>	<i>próximo</i>
-----------------------	-------------	----------------

O campo denominado *chave primária* é um número natural, que deverá contemplar números que estejam pelo menos no intervalo de 0 a (10^6-1) , incluindo valores extremos. Obviamente não é possível que haja repetição de valores nas diferentes entradas da tabela.

O campo denominado *dado* é simplesmente uma cadeia de caracteres (*string*) de comprimento variando de um a cinco, inclusive extremos. Ele deverá ser preenchido *aleatoriamente* à medida que novos elementos são inseridos na tabela.

O campo denominado *próximo* é um apontador, ou ponteiro, para o próximo elemento da tabela, já que esta está sendo implementada por meio de uma lista linear encadeada.

2. Como uma árvore binária de pesquisa (*binary search tree*) de número naturais, sendo que o tamanho da árvore deverá variar de forma idêntica à da lista linear encadeada.

A estrutura de cada *nó* (ou *nódulo*) da árvore é a seguinte:

<i>esquerda</i>	<i>chave primária</i>	<i>dado</i>	<i>direita</i>
-----------------	-----------------------	-------------	----------------

Os campos *chave primária* e *dado* são como aqueles definidos para a lista linear encadeada, enquanto que os campos *esquerda* e *direita* são apontadores, ou ponteiros, para as subárvores da esquerda e subárvore da direita, respectivamente. Lembrando que quaisquer umas destas subárvores pode ser, para um certo nó, *vazia*.

A aplicação desenvolvida deverá proporcionar as seguintes funcionalidades ao usuário:

1. escolher se utilizará a estrutura de dados *lista linear encadeada* ou *árvore binária de pesquisa* para a implementação da tabela.

A ação de escolha da estrutura de dados deverá ser sempre a primeira ação a ser realizada pelo usuário ao iniciar a aplicação, pois uma vez que ele escolha (*lista* ou *árvore*) não poderá alterá-la durante toda a execução da aplicação. Assim, se o usuário desejar trocar de implementação da tabela, deverá sair da aplicação e executá-la novamente;

2. definir o tamanho de tabela a ser utilizado naquele momento e que, sem deixar a aplicação, possa alterar este tamanho em qualquer momento que desejar¹;

3. inicializar, de forma aleatória, a tabela a ser utilizada. Se a tabela tiver até 100 elementos, deve haver a possibilidade de que o usuário forneça, *manualmente*, os elementos com os quais deseja inicializar a tabela. Tabela de maior tamanho somente podem ser inicializadas de maneira aleatória.

4. visualizar, na tela do computador, a tabela. Se a tabela tiver até 100 elementos, ela deverá ser apresentada *integralmente*.

Se a tabela tiver mais de 100 elementos e tiver sendo implementada por meio de uma lista linear encadeada, o usuário deverá fornecer o intervalo de posições dos elementos que deseja visualizar, sendo que este intervalo *nunca poderá ser maior que cem elementos*. Assim, o usuário poderá solicitar ver os elementos que estão nas posições de 10 a 40 (inclusive extremos), mas não poderá solicitar ver os elementos que estão nas posições de 100 a 1000 – já que o número de elementos solicitados é maior que 100 (901, para ser exato);

5. Se a tabela estiver sendo implementada por meio de uma lista linear encadeada, então as seguintes operações devem estar disponíveis:

- a) inserção de elemento. Serão, portanto, fornecidas a *chave primária* e o *dado* para o elemento a ser inserido;
- b) remoção de elemento. Sendo, portanto, fornecida a *chave primária* do elemento a ser removido;
- c) pesquisa sequencial ao elemento cuja *chave primária* for fornecida;
- d) pesquisa binária ao elemento cuja *chave primária* for fornecida²;

1 Os tamanhos possíveis de tabela variam de 10^1 a 10^6 : 10^1 , 10^2 , 10^3 , 10^4 , 10^5 ou 10^6 .

- e) ordenar a tabela em ordem crescente ou decrescente de *chaves primárias*, conforme escolha do usuário.
6. Se a tabela estiver sendo implementada por meio de uma árvore binária de pesquisa, então as seguintes operações devem estar disponíveis:
- a) inserção de elemento. Serão, portanto, fornecidas a *chave primária* e o *dado* para o elemento a ser inserido;
 - b) remoção de elemento. Sendo, portanto, fornecida a *chave primária* do elemento a ser removido;
 - c) pesquisa ao elemento cuja *chave primária* for fornecida.
7. registrar o horário em que a operação escolhida (qualquer uma delas) foi iniciada e terminada, bem como o tempo que consumiu – em segundos – bem como o tempo médio obtido por elemento. Por exemplo, supondo que a tabela tenha 100 elementos e que para pesquisar a chave primária igual a 246 tenha sido consumidos 2,3s, então o tempo médio desta operação é dado por $(2,3 / 100) = 0,023s$ – vinte e três milésimos de segundo.

IV. Da Entrega da Avaliação

A entrega desta AF deverá ser realizada até o final da aplicação da PRIMEIRA AVALIAÇÃO FORMAL da disciplina, que está prevista para a noite de 26 de maio de 2016, quinta-feira. Havendo alteração na data de aplicação da avaliação formal, automaticamente se a data de entrega desta AF.

A AF deverá ser entregue por meio de seu armazenamento em mídia DVD-ROM (obrigatoriamente), contendo:

1. Todos os programas-fonte, programas executáveis e bibliotecas de suporte e/ou complementares necessários para o pleno funcionamento/teste da aplicação pelo docente responsável pela disciplina³. Se a linguagem de programação utilizada for:
 - a) JAVA: a aplicação deve ser entregue por meio de um arquivo .JAR ou pacote de instalação;
 - b) C/C++: deve ser explicitado qual o ambiente operacional utilizado (Windows, Linux, Mac OS, ...) para a elaboração da aplicação, incluindo-se a versão do compilador utilizado para sua produção;
 - c) Outra linguagem e/ou ambiente: detalhar, por meio de um arquivo denominado **LINGUAGEM.TXT**, quais os requisitos necessários para que a aplicação seja testada.
2. Quaisquer *softwares* adicionais que devam ser instalados no equipamento para que se possa testar a aplicação;
3. Um arquivo denominado **INTEGRANTES.TXT** que contenha o nome completo e número de matrícula de cada um dos integrantes da equipe de acadêmicos que elaborou o trabalho. Um acadêmico deve ser identificado como *líder da equipe* e fornecido seu telefone celular para contato;
4. Se a equipe considerar que são necessárias *instruções adicionais* para que o docente consiga testar a aplicação desenvolvida, estas devem ser registradas no arquivo denominado **LEIAME.TXT** – que deverá ser armazenado na *raiz* do DVD-ROM entregue.

Observações:

- a) Não serão aceitas AFs entregues em *pendrives*, sob qualquer hipótese e/ou alegação;
- b) Recomenda-se que a equipe armazene um DVD-ROM como *cópia de segurança (backup)* da AF entregue;
- c) O DVD-ROM contendo a AF deverá estar com capa (de papel ou de plástico) e etiqueta autoadesiva. Na etiqueta deverá constar o *nome de cada um dos integrantes da equipe*, conforme modelo a seguir:

2 Nesta situação a própria aplicação deverá se a tabela está com suas chaves previamente *ordenadas* (de forma crescente ou decrescente). Se não estiver, deve solicitar ao usuário que primeiro realize a operação de ordenação para que, somente depois de sua realização, a pesquisa binária possa ser efetivada.

3 O equipamento utilizado pelo docente responsável para realizar o teste/avaliação da aplicação entregue pela equipe de acadêmicos possui a seguinte configuração e ambiente operacional: microcomputador com processador Intel i7 de 3.4GHz de velocidade, 32GB de memória principal, placa de vídeo Nvidia GeForce de 4GB (o monitor, portanto, atinge a resolução *FULL HD* – 1920 x 1080 pixels) e SSD de 480GB. O sistema operacional é o Linux Ubuntu 14.04 LTS, mantido constantemente atualizado.

Bacharelado em Engenharia de Software
AOB – Avaliação Formativa
1º Bimestre – 2016/1

Gastão	- 201510101
Huguinho	- 201594393
Luizinho	- 201594949
Pato Donald	- 201594944
Tio Patinhas	- 201509809
Zezinho	- 201598237

V. Do que Será Avaliado

Durante o processo de atribuição de *nota* à AF serão considerados os seguintes critérios:

1. O correto funcionamento das operações e funcionalidades solicitadas (inclusão, exclusão, pesquisa, etc.);
2. Da *usabilidade* de aplicação desenvolvida, ou seja, a medida da facilidade que um usuário tem para utilizar corretamente a aplicação desenvolvida, realizando as operações desejadas;
3. Da qualidade do código-fonte produzido (clareza, facilidade de leitura devido à utilização de técnicas adequadas de programação e documentação, organização do código-fonte, etc.).

VI. Das Dúvidas

As dúvidas quanto ao desenvolvimento da aplicação especificada poderão ser sanadas por meio de:

1. Consultas ao docente responsável, em sala de aula, durante as aulas presenciais;
2. Submissão de questionamento no fórum denominado “Dúvidas e/ou Questões sobre a PRIMEIRA avaliação formativa”. Este fórum está disponível na área da disciplina na Plataforma de Educação a Distância do INF/UFG, em <https://ead.inf.ufg.br>.