Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II
Professor: Mário Luiz Rodrigues Oliveira
Atividade: 2ª Trabalho Prático
Formiga, MG, 13 de janeiro de 2015

INSTRUÇÕES:

1. **Esta atividade poderá ser resolvida em grupo com no máximo 2 integrantes.**
2. **Caso você ache que falta algum detalhe nas especificações, você deverá fazer as suposições que julgar necessárias e escrevê-las no seu relatório. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.**
3. **Como produtos da atividade serão gerados dois artefatos: códigos fontes da implementação e documentação da atividade.**
4. **Cada arquivo-fonte deve ter um cabeçalho constando as seguintes informações: nome(s) do(s) aluno(s), matrícula(s) e data.**
5. **O arquivo contendo a documentação da atividade (relatório) deve ser devidamente identificado com o(s) nome(s) e matrícula do(s) autor(es) do trabalho. O arquivo contendo o relatório deve, obrigatoriamente, estar no formado PDF.**
6. **Devem ser entregues os arquivos contendo os códigos-fontes e o arquivo contendo a documentação da atividade (relatório). Compacte todos os artefatos gerados num único arquivo no formato RAR.**
7. **Entregue apenas uma resolução por grupo. A atividade deve ser entregue via portal acadêmico acessado pela URL:https://meu.ifmg.edu.br/.**
8. **O prazo final para entrega desta atividade é até 23:59:00 do dia 02/02/2015.**
9. **O envio é de total responsabilidade do aluno. Não serão aceitos trabalhos enviados fora do prazo estabelecido.**
10. **Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos.**
11. **O valor desta atividade é 15 pontos.**

Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II
Professor: Mário Luiz Rodrigues Oliveira
Atividade: 2ª Trabalho Prático
Formiga, MG, 13 de janeiro de 2015

1. Objetivos e Descrição do Trabalho

Esse trabalho possui dois objetivos. O primeiro objetivo deste trabalho é realizar um estudo empírico sobre comportamento da altura de uma árvore binária de busca na qual são feitas inserções e remoções de forma aleatória. O segundo objeto desse trabalho é a implementação de uma aplicação que identifique erros ortográficos num arquivo texto.

Para cumprir o primeiro objetivo realize um experimento aplicando inserção e remoção de elementos aleatórios em uma árvore binária de busca aleatoriamente criada. Faça medidas da altura da árvore binária de busca nas seguintes condições:

- crie aleatoriamente uma árvore binária de busca com mil nós e calcule a altura desta árvore. Realize aleatoriamente 1.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos 100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;
- crie aleatoriamente uma árvore binária de busca com 5 mil nós e calcule a altura desta árvore. Realize aleatoriamente 5.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos 100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;
- crie aleatoriamente uma árvore binária de busca com 10 mil nós e calcule a altura desta árvore. Realize aleatoriamente 10.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos 100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;
- crie aleatoriamente uma árvore binária de busca com 16 mil nós e calcule a altura desta árvore. Realize aleatoriamente 16.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos 100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;
- crie aleatoriamente uma árvore binária de busca com 32 mil nós e calcule a altura desta árvore. Realize aleatoriamente 32.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos

100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;

- crie aleatoriamente uma árvore binária de busca com 50 mil nós e calcule a altura desta árvore. Realize aleatoriamente 50.000 operações de inserção/remoção de números escolhidos aleatoriamente e após essas operações calcule a quantidade de nós presentes na árvore e a sua nova altura. Repita esses experimentos 100 vezes e ao final reporte a altura média da árvore inicialmente criada e a altura média e quantidade média de nós na árvore após as operações de inserção/remoção;

Para atingir o segundo objetivo faça uma aplicação que receba os seguintes dados:

- um arquivo texto referente ao dicionário;
- um arquivo referente ao arquivo a ser analisado

No final sua aplicação deve reportar a quantidade de erros encontrados e a(s) linha(s) e coluna(s) onde há palavras grafadas incorretamente. O valor da coluna deve indicar a posição do primeiro caracter de uma palavra grafada incorretamente. Considere o caracter espaço em branco como delimitador de palavras e como palavra quaisquer sequência composta unicamente por letras e que possua no mínimo 4 letras. E considere como letras, as letras definidas no alfabeto da língua portuguesa. As palavras constantes no arquivo a ser analisado e inexistentes no dicionário serão reportadas como erros ortográficos. Um ponto crucial para a sua aplicação ser eficiente em termos de tempo de processamento é a escolha da estrutura de dados usada para representar o dicionário. Nesse trabalho use uma árvore binária de busca para armazenar as palavras pertencentes ao dicionário.

Ademais, o sistema projetado deve também atender as seguintes especificações:

- o código fonte do programa deve ser portável, ou seja, o mesmo código fonte deve ser compilado corretamente e gerar código executável paras as seguintes plataformas: sistemas operacionais Windows e GNU/Linux e arquiteturas *Intel* x86/x86-64 e *AMD*64/x86-64;
- ser implementado, exclusivamente, na linguagem de programação C padrão *ANSI;*
- o código fonte do programa deve ser compatível com o compilador *GNU Compiler Collection* (GCC versão 4.6.1 ou superior). Para verificar a versão do compilador GCC instalado no seu computador use a opção –version. Adicionalmente o código deve ser compatível com as seguintes opções de compilação do GCC:
    1. -Wall: ativa todos avisos de compilação;
    2. -pedantic -ansi: compila o programa estritamente de acordo com o padrão ANSI.

Na seção 2 são descritos os artefatos e o que deve ser entregue como produto desta atividade.

2. Artefatos

Esta seção descreve o que deve ser gerado como produto final do trabalho. Ao final do trabalho deve ser gerado além das implementações um relatório documentando seu sistema, com as seguintes informações:

1. introdução: descrever o problema resolvido e apresentar uma visão geral do sistema implementado;
2. implementação: descrição sobre as decisões de projeto e implementação do programa. Essa parte da documentação deve incluir uma descrição de forma sucinta das estruturas de dados usadas no programa; descrição dos tipos abstratos de dados (TADs) definidos, explicitando as operações disponíveis; explicações de como os TADs são implementados; funcionamento das principais funções e procedimentos utilizados; o formato de entrada e saída dos dados, como executar o programa e as decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado;
3. descrição dos testes realizados;
4. conclusão: avaliação do grupo sobre o trabalho considerando a experiência adquirida, a contribuição para o aprendizado da disciplina, as principais dificuldades encontradas ao implementá-lo e como tais dificuldades foram superadas;
5. bibliografia: cite as fontes consultadas na resolução do trabalho.

Todos os artefatos (códigos fontes e relatório) devem ser entregue conforme as instruções contidas nesse documento.

Na seção 3 são descritos os critérios de correção desta atividade.

3. Critérios de Correção

Conforme descrito no plano de ensino, os critérios de avaliação do trabalho são:

1. somente serão corrigidos os trabalhos com códigos fontes portáveis e sem de erros de compilação;
2. análise código fonte: modularização, uso adequado de comentários, legibilidade e edentação do código; (20%)
3. apresentação do trabalho e execução correta numa bateria de testes práticos; (40%)
4. uso adequado de TAD e estruturas de dados; (20%) e
5. documentação. (20%)

Na seção 4 indica-se a bibliografia consultada para a confecção deste documento.

4. Bibliografia

Drozdek, Adam. Estruturas de Dados e Algoritmos em C++. São Paulo: Cengage Learning, 2010.

Edelweiss, Nina; Galante, Renata. Estruturas de Dados. Porto Alegre: Bookman, 2009.