



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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

INSTRUÇÕES:

1. Esta atividade deve, obrigatoriamente, ser resolvida individualmente.
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 três artefatos: códigos fontes da implementação, arquivo *MakeFile* e documentação da atividade.
4. Cada arquivo-fonte deve ter um cabeçalho constando as seguintes informações: nome autor(a), matrícula e data.
5. O arquivo contendo a documentação da atividade (relatório) deve ser devidamente identificado com o nome e matrícula do(a) autor(a) do trabalho. O arquivo contendo o relatório deve, obrigatoriamente, estar no formato **PDF**.
6. Devem ser entregues os arquivos contendo os códigos-fontes, arquivo *MakeFile* e o arquivo contendo a documentação da atividade (relatório). Compacte todos os artefatos gerados **num único arquivo no formato RAR**.
7. A atividade deve ser entregue, obrigatoriamente, 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 **03/10/2016**.
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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

1. Índice Remissivo

Segundo Ziviani (2011), várias aplicações necessitam de um relatório de referências cruzadas. Por exemplo, a maioria dos livros apresenta um índice remissivo que corresponde a uma lista alfabética de palavras-chaves ou palavras relevantes do texto com a indicação das linhas no texto onde cada palavra-chave ocorre. Na verdade, o índice remissivo é um arquivo invertido¹. Os arquivos invertidos constituem um importante mecanismo de indexação e são muito utilizados nas máquinas de busca na *Web*.

Como exemplo, suponha um arquivo contendo um texto constituído por:

1. *Good programming is not learned from*
2. *generalities, but by seeing how significant*
3. *programs can be made clean, easy to*
4. *read, easy to maintain and modify,*
5. *human-engineered, efficient, and reliable,*
6. *by the application of common sense and*
7. *by the use of good programming practices.*

Assumindo que o índice remissivo seja constituído das seguintes palavras-chave: *programming, programs, easy, by, human-engineered, and, be, to*, o sistema para a criação do índice deve produzir a seguinte saída:

<i>and</i>	4	5	6
<i>be</i>	3		
<i>by</i>	2	6	7
<i>easy</i>	3	4	
<i>human-engineered</i>	5		
<i>programming</i>	1	7	
<i>programs</i>	3		
<i>to</i>	3	4	

Note que a lista de palavras-chave está em ordem alfabética. Adjacente a cada palavra está uma lista de números de linhas, indicando as linhas onde cada palavra ocorre no texto.

¹Um arquivo invertido é constituído de duas partes: vocabulário e ocorrências. O vocabulário é o conjunto de todas as palavras distintas no texto. Para cada palavra distinta, uma lista de posições onde ela ocorre no texto é armazenada. O conjunto das listas é chamado de ocorrências. As posições podem referir-se a palavras ou caracteres.



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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

As seções seguintes descrevem o trabalho e seus objetivos, os artefatos que devem ser gerados ao final do trabalho, os critérios de correção do trabalho e finalmente a bibliografia consultada.

2. Objetivos e Descrição do Trabalho

O objetivo geral deste trabalho é o projeto e a implementação de um sistema que construa um índice remissivo. E o objetivo específico é exercitar o conceito e aplicações das estruturas de dados tabela *hash* e lista encadeada realizando uma comparação entre essas duas estruturas de dados quando utilizadas na implementação de um índice remissivo.

Para criar o índice remissivo, o sistema deverá ler um texto contendo um número arbitrário de palavras-chave que deverão constituir o índice remissivo, seguido da leitura de um texto de tamanho arbitrário, o qual deverá ser escaneado à procura de palavras-chave que pertençam ao índice remissivo. Como saída o sistema deverá gerar um arquivo texto contendo o índice remissivo (referências cruzadas). O nome desses 3 arquivos deverão ser informados ao sistema como parâmetros na linha de comando

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

- o código fonte do programa deve ser portátil, ou seja, o mesmo código fonte deve ser compilado corretamente e gerar código executável para as seguintes plataformas: sistemas operacionais Windows e GNU/Linux e arquiteturas *Intel* x86/x86-64 e *AMD64*/x86-64;
- ser implementado, exclusivamente, na linguagem de programação C;
- o código fonte do programa deve ser compatível com o compilador *GNU Compiler Collection* (GCC versão 4.6.1);
- fornecer um arquivo *MakeFile* que permita compilar adequadamente os códigos-fontes do programa;
- utilizar as estruturas de dados tabela *hash* e lista encadeada para implementar o índice remissivo;
- usar tanto o método *open hash* quanto encadeamento exterior para resolver colisões;
- deve-se implementar e comparar 3 opções do método *open hash*. O método *hashing* linear é obrigatório e outros dois métodos devem ser implementados a critério do(a) autor(a) do trabalho;
- o(a) autor(a) do trabalho deve determinar o tamanho adequado da tabela *hash*. Para tanto o(a) autor(a) deve testar 3 tamanhos diferentes para tabela *hash*;



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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

- o(a) autor(a) do trabalho, também, deve determinar a função *hash* adequada. Para tanto o(a) autor(a) deve testar 3 funções *hash* diferentes. O uso da função *hash* sugerida na referência Ziviani (Ziviani, 2011) é obrigatório. As outras duas funções é a critério do(a) autor(a) do trabalho;
- considerar uma palavra como uma sequência de letras e dígitos, começando com uma letra;
- apenas os primeiros **c1** caracteres devem ser retidos nas chaves. Assim, duas palavras que não diferem nos primeiros **c1** caracteres são consideradas idênticas;
- palavras constituídas por menos do que **c1** caracteres devem ser preenchidas por um número apropriado de brancos.
- para esse trabalho considere **c1**=32 caracteres;
- sequências que iniciam-se com letras e possuem menos de 4 caracteres não serão consideradas palavras; e
- o arquivo contendo o índice remissivo deve ter seu conteúdo disposto em ordem alfabética;
- ao final da implementação do trabalho o(a) autor(a) do trabalho deverá responder adequadamente qual estrutura de dados é a mais indicada para os objetivos aqui descritos. Deve, de igual forma, responder qual o método de tratamento de colisão mais adequado, qual o tamanho ideal para a tabela *hash* e qual a função *hash* mais indicada para os propósitos desse trabalho.

Na seção 3 são descritos os artefatos e o que deve ser entregue como produto desta atividade.

3. Artefatos

Esta seção descreve o que deve ser gerado como produto final do trabalho. Ao final do trabalho deve ser gerado além das implementações e do arquivo *MakeFile*, 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



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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

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. estudo da complexidade: estudo da complexidade (notação O) do tempo de execução dos métodos implementados e do programa como um todo;
5. análise dos resultados: análise os resultados encontrados nos testes realizados. Procure responder os seguintes questionamentos:
 - Qual das estruturas de dados testadas é indicada para implementar de forma eficiente um índice remissivo?
 - Qual o tamanho indicado para a estrutura de dados tabela *hash*?
 - Qual a função *hash* apresentou melhores resultados?
 - Qual método de tratamento de colisão apresentou melhores resultados?
6. conclusão: recomendações do(a) autor(a) a respeito das análises feitas e sobre os questionamentos mencionados no item 5 dessa seção. Avaliação do(a) autor(a) sobre a contribuição desse trabalho para o aprendizado da disciplina;
7. bibliografia: cite as fontes consultadas na resolução do trabalho;
8. listagem dos códigos-fontes do programa e do arquivo *MakeFile*.

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

Na seção 4 são descritos os critérios de correção desta atividade.

4. 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 portáteis e sem erros de compilação;
2. análise código fonte: modularização, uso adequado de comentários, legibilidade e indentação do código; (10%)
3. uso adequado de TAD e estruturas de dados; e (20%)
4. execução correta numa bateria de testes práticos; (50%)
5. documentação. (20%)



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: 1ª Trabalho Prático
Formiga, MG, 13 de setembro de 2016

6. A critério do professor poderá ser agendada uma apresentação oral do trabalho. Na ausência de plágio, as notas dos trabalhos serão computadas da seguinte forma: $\text{nota_apresentação} * \text{nota_trabalho}$, ou seja, a nota final é ponderada pela nota da apresentação.

Na seção 5 indica-se a bibliografia consultada para a confecção deste documento.

5. Bibliografia

Ziviani, Nívio. Projeto de Algoritmos: com implementações em Pascal e C. 3ª edição revista e ampliada. São Paulo: Cengage Learning, 2011.