



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO

Disciplina: Estrutura de Dados I	Turma: EC3/CC4	Data: 03/06/15	Nota:
Professor: Renato E. N. de Moraes	Semestre: 2015-1	Valor: 3,0 pt	
Aluno:	Trabalho 02 - Árvores		

O objetivo deste trabalho é utilizar o algoritmo de compactação de mensagens de Huffman para criar arquivos binários (ou texto) compactados.

O algoritmo de Huffman sugere um esquema de codificação para o alfabeto de uma mensagem em função da frequência de cada símbolo. Essa codificação é representada através de uma árvore binária. Cabe a você pesquisar a literatura e entender a forma de criar essa árvore e codificar a mensagem.

Uma vez entendido o algoritmo de Huffman, a mensagem (arquivo texto) deverá ser codificada e armazenada em arquivo. No entanto, antes é recomendado que a árvore que gera a codificação seja armazenada no cabeçalho do arquivo para posterior descompactação. Lembre-se que o tamanho da árvore é variável – ela depende do tamanho do alfabeto. Cabe a você desenvolver o TAD para esta árvore e a forma de armazenamento. Ressalta-se que essa é a parte principal do trabalho e cópias serão severamente punidas.

Você também deverá implementar uma função para descompactar o arquivo. Com essa segunda função implementada, você poderá facilmente testar se o seu trabalho prático está funcionando perfeitamente. O seu programa deverá funcionar em linha de comando. Deseja-se também que o tamanho dos arquivos compactados pelo algoritmo de Huffman sejam comparados com os gerados por compactadores bem conhecidos como ZIP, RAR e TAR. Uma rápida análise dessa comparação deve ser feita.

Procure utilizar arquivos texto de diversos tamanhos, variando desde 1 Kilobytes até 100 Megabytes.

**O que deve ser entregue.** Entre outras coisas, a documentação deve conter:

- Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
- Implementação: descrição sobre a implementação dos algoritmos, incluindo o algoritmo de Huffman. Deve ser detalhada a estrutura de dados utilizada (excencialmente com diagramas ilustrativos), o funcionamento dos algoritmos e decisões tomadas relativas aos detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados na implementação NÃO devem ser inseridos na documentação.
- Análise dos testes: analisar os tamanhos dos arquivos compactados a partir dos experimentos realizados.
- Conclusão: comentários gerais sobre quais algoritmos são os mais eficientes.
- Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização.
- Formato: deve ser PDF

O Trabalho deve ser feito em grupos de até 3 alunos. O programa funcionando deverá ser apresentado para o professor em uma máquina no laboratório no dia 24/06/2015 (não prorrogável), segundo agendamento prévio. A apresentação será feita por um único aluno a ser sorteado entre os alunos presentes no momento da apresentação. Alunos ausentes no momento da apresentação receberão nota zero. **Não haverá em hipótese alguma outro momento de apresentação.** Os arquivos de entrada serão fornecidos pelo autor do trabalho no dia da apresentação. O professor poderá também fornecer arquivo próprio. Quando solicitado, o autor deverá explicar o código e justificar os resultados obtidos.

Comentários Gerais:



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**

---

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, TAD's, identificação e comentários no programa também vão valer pontos.
- Você **NÃO** deve utilizar estruturas de dados já implementadas na linguagem tais como `array_list`, `linked_list` ou similares. Ou seja, você deverá implementar todas as estruturas utilizadas.