Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas
Programa de Pós-Graduação em Ciência da Computação
Projeto e Análise de Algoritmos - 1º semestre de 2010
Professor: David Menotti

Trabalho Prático 5
Processamento de Cadeias de Caracteres
Algoritmos Exatos

Valor: 1,0 pontos (10% da nota total) Data de Entrega: 14/07/2010

Descrição

Uma cadeia de caracteres é uma sequência qualquer de elementos. As cadeias aparecem no processamento de textos em linguagem natural, códigos, dicionários, sequenciamento de DNA em biologia computacional, representação de imagens por meio de *bitmaps*, dentre outros [2]. Neste trabalho serão estudados algoritmos de uma classe de problemas que envolvem a manipulação de cadeias, mais particularmente algoritmos que estão relacionados com pesquisa exata.

Uma cadeia corresponde a uma sequência de elemtnos denomiados caracteres. Os caractes são escolhdios de um conjunto denominado **alfabeto**. Por exemplo, em uma cadeia de *bits* o alfabeto é $\{0, 1\}$. A pesquisa em cadeias de caracteres é um componente importante em diversos problemas computacionais, tais como edição de texto, recuperação de informação e estudo de sequências de DNA em biologia computacional. No caso de programas editores de texto, o usuário pode estar interessado em buscar todas as ocorrências de um padrão (uma palavra particular) no texto que está sendo editado. Este problema é conhecimento como casamento de cadeias de caracteres ou casamento de padrão (do inglês *pattern matching*).

O problema de **casamento de cadeias** ou **casamento de padrão** pode ser formalizado da seguinte forma. O texto é um arranjo $T[1..n]$ de tamanho n e o padrão é um arranjo $P[1..m]$ de tamanho $m \leq\leq n$. Os elementos P e T são escolhidos a partir de um alfabeto finito Σ de tamanho c. Por exemplo, podemos ter $\Sigma = \{0, 1\}$ ou $\Sigma = \{a, b, \ldots, z\}$. Dadas duas cadeias P (padrão) de comprimento $|P| = m$ e T (texto) de comprimento $|T| = n$, em que $n >> m$, deseja-se saber as ocorrências de P em T.

Os algoritmos para casamento de cadeias podem ser classificados em três categorias maiores:

- Padrão e texto não pré-processados;
- Padrão pré-processado;

- Padrão e texto pré-processados.

Este trabalho concentra-se na segunda categoria, ou seja, algoritmos de casamento de padrão com padrão pré-processado. Esses algoritmos são do tipo sequêncial e o padrão é conhecido anteriormente, o que permite o seu pré-processamento. Os algoritmos têm complexidade de tempo $O(n)$ e complexidade de espaço $O(m+c)$, no pior caso. Os algoritmos nesta categoria são usados em diversas aplicações como, por exemplo, em programas para edição de textos.

O trabalho prático 5 consiste em implementar 4 algoritmos de processamento/-casamento de cadeias de caracteres[1, 3]

- FB - Algoritmo de Força Bruta;
- BMH - Algoritmo de Boyer-Moore-Horspool
- BMHS - Algoritmo de Boyer-Moore-Horspool-Sunday
- SAE - Algoritmo *Shift-And* Exato

por equipe (veja Tabela 1 - o grupo formado por um único aluno deve implementar somente dois algoritmos).

As implementações deverão ser testadas e analisadas com vários arquivos (T) e palavras (P). A escolha dos arquivos para testes, e o conjunto de padrões (no mínimo 20- no máximo 30) é tarefa da equipe. Devem ser utilizados arquivos de vários tamanho (em potências de 10), visando a obtenção de tempos viáveis para a comparação das implementações. Lembrem-se que essas escolhas fazem parte da avaliação. E maior enfase na correção será dada a análise dos resultados.

O que deve ser entregue

- Código fonte dos programas em C ou C++ (bem identado e comentado).
- Documentação do trabalho.

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição dos algoritmos estudados e implementados.
2. Implementação: descrição sobre as implementações. Muito importante: os códigos utilizados nas implementações devem ser inseridos na documentação.
3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementadas.
4. Análise de Resultados: comparar os dados obtidos e discussar sobre estes.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sítio da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto onde é utilizada.

7. Em LaTeX: A documentação deve ser elaborada obrigatoriamente em LaTeX. Veja modelo de como fazer o trabalho em latex: `http://www.decom.ufop.br/menotti/paa101/tps/modelo.zip`

8. Formato final: mandatoriamente em PDF (`http://www.pdf995.com/`).

No.	Alunos
1	RENSSO VICTOR ANGÉLICA APARECIDA
2	ALAN ROBERT FELIPE SANTIAGO
3	MATHEUS NOHRA THAIS COTTA
4	SAMUEL EVANGELISTA TARIK DE MELO
5	ANDRÉ SIQUEIRA SAUL EMANUEL
6	JEAN CARLO LARISSA NATÁLIA
7	PEDRO HENRIQUE JOÃO TACIO
8	FABRÍCIO
9	JADER LUIZ
10	RONAN LOSCHI

Tabela 1: Alocação das duplas.

Como deve ser feita a entrega

A entrega DEVE ser feita via Moodle (`www.decom.ufop.br/moodle`) na forma de um único arquivo zipado, contendo o código fonte, arquivos diversos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho.

Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- O trabalho é em dupla (grupo de DOIS alunos) (veja Tabela 1);
- Trabalhos copiados (e FONTE) terão nota zero. Devido a recorrentes problemas com cópias de trabalhos (plágios), os autores de trabalhos copiados também terão a maior nota dentre os testes teóricos levada a zero, como forma de punição e coação ao plágio acadêmico;

- Trabalhos entregues em atraso terão descontados 0,1 pontos por hora;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.

Referências

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press/McGraw-Hill, 3rd edition, 2009.

[2] N. Ziviani. *Projeto de Algoritmos: com implementações em Pascal e C*. Cengage Learning (Thomson / Pioneira), São Paulo, 2nd edition, 2004.

[3] N. Ziviani. *Projeto de Algoritmos com implementações em Java e C++*. Cengage Learning (Thomson / Pioneira), São Paulo, 1st edition, 2006.