Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas
Departamento de Computação

CASAMENTO DE CADEIAS

Axel Junio Almeida de Andrade 15.1.4220
Mateus de Paula Abreu 15.1.4022

Prof.: Dr. Guilherme Tavares Assis

Ouro Preto
2017

1. Introdução

Textos estão presentes em diversos locais, como livros, revistas e etc, e são o principal meio de comunicação após a linguagem falada. É comum que se queira localizar uma palavra ou frase, por exemplo, dentro de um texto e, dependendo do tamanho do texto, se torna uma tarefa muito cansativa e suscetível a erro quando executada por humanos.

Para isso, existem diversos algoritmos que executam o processo computadorizado de localização de padrões dentro de um texto, chamado de casamento de cadeias. A maioria das linguagens de programação modernas já possuem bibliotecas com funções para lidar com cadeias, incluindo uma ou mais funções de pesquisa de padrão. Entretanto, é interessante entender os princípios destes algoritmos e fazer uma análise experimental.

2. Métodos de casamento exato de cadeias

O casamento exato de um padrão em um texto se dá quando todos os caracteres do padrão são encontrados no texto exatamente na ordem em que os mesmos ocorrem no padrão.

Uma busca exata por uma palavra qualquer poderia ser feita e um casamento ocorreria se a palavra fosse encontrada no texto original. No entanto, se no texto a palavra buscada estivesse sido escrita de forma errada, não teríamos casamento algum.

Nas próximas seções, apresentaremos quatro algoritmos para casamento exato de padrão que foram utilizados para a realização deste trabalho: Força Bruta; Boyer-Moore; Boyer-Moore-Horspool, Boyer-Moore-Horspool-Sunday e Shift-And Exato.Para os testes, utilizamos textos dos tamanhos: pequeno, médio e grande.

Para gerá-los, utilizamos o gerador http://br.lipsum.com/, bastante utilizado para preencher espaços antes de que o texto oficial esteja disponível. Os tamanhos dos textos, de acordo com o que foi pedido, são:

Pequeno: $3.000 \leq n \leq 5.000$

Médio: $220.000 \leq n \leq 250.000$

Grande: $1.700.000 \leq n \leq 2.000.000$

Para cada texto T de tamanho n, buscamos ocorrências de 3 padrões P dos seguintes tamanhos:

Pequeno: $3 \leq m \leq 5$

Médio: $10 \leq m \leq 15$

Grande: $25 \leq m \leq 30$

Utilizamos a linha ./casamento <método> <texto> <padrão> <-P> para executar, onde:

Método: 1. Força Bruta, 2. Boyer-Moore, 3. Boyer-Moore-Horspool, 4. Boyer-Moore-Horspool-Sunday, 5.Shift-And

Texto: “pequeno.txt”, “medio.txt”, “grande.txt”

Padrão: “amet”, “Loremipsum”, “Suspendissetristiqueipsum”

Os testes foram executados no Sistema Operacional Linux versão Ubuntu, Intel® Core™ i7-4500U CPU 1.80GHz × 4 com 8gb de RAM.

2. 1 Força Bruta

Mais simples de todos, deve verificar cada caractere do padrão com os m primeiros caracteres do texto, onde m é o tamanho do padrão. A ideia consiste em tentar casar qualquer subcadeia de comprimento m no texto com o padrão desejado.

TABELA FORCA BRUTA

Tamanho do texto	Tamanho do padrão	Deslocamento	Comparações	Tempo (s)
PEQUENO	PEQUENO	0	29916	0.000454
PEQUENO	MEDIO	0	28864	0.000471
PEQUENO	GRANDE	0	29030	0.000127
MEDIO	PEQUENO	0	20889	0.000423
MEDIO	MEDIO	0	20035	0.000238
MEDIO	GRANDE	0	20156	0.000256
GRANDE	PEQUENO	0	5269703	0.043992
GRANDE	MEDIO	0	5051644	0.028533
GRANDE	GRANDE	0	5071990	0.011692

Com os testes acima, podemos observar que a força bruta faz muitas comparações, o que já era de se esperar pelo funcionamento do método. Mesmo que as comparações altas possam fazer com que seja um algoritmo lento. Isto pode ser justificado pelos textos relativamente pequenos e pelos processadores utilizados nos testes.

2.2 Boyer-Moore (heurística por ocorrência)

O algoritmo Boyer-Moore para casamento de cadeias é considerado como o mais eficiente para aplicações simples. Muito utilizado por editores de textonas ferramentas localizar e substituir. Ele busca pelos caracteres do padrão no texto da direita para esquerda e, em caso de colisão (não casamento) utiliza uma heurística para ignorar os próximos t caracteres do texto e continuar a busca a partir deste ponto.

Essa heurística pode ser: por ocorrência e por casamento. Na heurística por ocorrência, andamos para a direita até que o caractere do texto que causou colisão case com algum caractere do padrão. Na heurística por casamento, andamos para direita até que case com o pedaço do texto casado anteriormente.

TABELA BOYER MOORE

Tamanho do texto	Tamanho do padrão	Deslocamento	Comparações	Tempo (s)
PEQUENO	PEQUENO	14428	23099	0.000363
PEQUENO	MEDIO	14436	18512	0.000126
PEQUENO	GRANDE	14427	16528	0.000231
MEDIO	PEQUENO	10013	16052	0.000451
MEDIO	MEDIO	10019	12850	0.000204
MEDIO	GRANDE	10034	11510	0.000248
GRANDE	PEQUENO	2523617	4048089	0.036781
GRANDE	MEDIO	2523618	3231799	0.024637
GRANDE	GRANDE	2523632	2876184	0.013166

2.3 Boyer-Moore-Horspool

Versão do Boyer-Moore que possui como atualização realizar um pré-processamento sobre o texto, obtendo uma tabela com valores de deslocamento para cada caractere. Este valor de deslocamento é utilizado para “andar” no texto em caso de colisão.

TABELA BMH

Tamanho do texto	Tamanho do padrão	Deslocamento	Comparações	Tempo (s)
PEQUENO	PEQUENO	3992	4500	0.000369
PEQUENO	MEDIO	1964	2175	0.000147
PEQUENO	GRANDE	914	1084	8.8e-05
MEDIO	PEQUENO	2789	3156	0.000291
MEDIO	MEDIO	1324	1487	0.000175
MEDIO	GRANDE	646	763	8.5e-05
GRANDE	PEQUENO	704386	799252	0.030912
GRANDE	MEDIO	335296	368419	0.00652
GRANDE	GRANDE	161686	165906	0.004269

O pré-processamento do padrão ocorre no começo do código. Neste pré-processamento é calculado a tabela de deslocamento d. A pesquisa é realizada através de dois loops alinhados. O loop mais externo varia entre m e n com incrementos equivalentes ao endereço da tabela de deslocamento do caractere que está na i+1-ésima posição no texto, a qual corresponde à posição do último caractere de P.

O deslocamento da ocorrência pode ser pré-computado com base apenas no padrão e no alfabeto, e a complexidade de tempo e espaço para esta fase é O(c), onde c é o tamanho do alfabeto. Para a fase de pesquisa, o pior caso do algoritmo é O(nm), caso os deslocamentos ocorram apenas de 1 em 1. No entanto o melhor caso é O(n /m), onde os descolamentos são maiores, do tamanho do padrão.

2.4 Boyer-Moore-Horspool-Sunday

Sunday apresentou outra simplificação importante para o algoritmo BM, ficando conhecida como BMHS, que é uma variante do BMH. Ao endereçar a tabela com o caractere no texto correspondente ao próximo caractere após o último caractere do padrão, em vez de deslocar o padrão usando o último caractere como no algoritmo BMH.

Para pré-computar o padrão, o valor inicial de todas as entradas na tabela de deslocamentos é feito igual a m + 1. A seguir, os m primeiros caracteres do padrão são usados para obter os outros valores da tabela.

TABELA BMHS

Tamanho do texto	Tamanho do padrão	Deslocamento	Comparações	Tempo (s)
PEQUENO	PEQUENO	3352	3998	0.00029
PEQUENO	MEDIO	1752	2035	8.7e-05
PEQUENO	GRANDE	910	1112	4.7e-05
MEDIO	PEQUENO	2285	2765	0.000339
MEDIO	MEDIO	1199	1384	4e-05
MEDIO	GRANDE	644	816	4.3e-05
GRANDE	PEQUENO	574804	693715	0.023256
GRANDE	MEDIO	302169	339727	0.008417
GRANDE	GRANDE	164257	186097	0.004978

O comportamento do BMHS é igual ao do algoritmo BMH. Porém os deslocamentos são mais maiores, com alguns saltos relativamente mais longos em padrões menores. Por exemplo, para exemplo, para um padrão de tamanho m = 1, o deslocamento é igual a 2m quando não há casamento.

2.4 Shift-And

O algoritmo Shift-And utilizada o conceito de paralelismo de bit, uma técnica que tira proveito do paralelismo intrínseco das operações sobre bits dentro de uma palavra de computador.O algoritmo faz uso de um autômato para representar o padrão a ser procurado. Este autômato representa uma transição de estados, que ocorre quando um determinado caractere do padrão é encontrado no texto.

TABELA SHIFT AND

Tamanho do texto	Tamanho do padrão	Deslocamento	Comparações	Tempo (s)
PEQUENO	PEQUENO	0	14426	0.000291
PEQUENO	MEDIO	0	14426	0.000124
PEQUENO	GRANDE	0	14426	0.000370
MEDIO	PEQUENO	0	10009	0.000466
MEDIO	MEDIO	0	10009	0.00024
MEDIO	GRANDE	0	10009	0.000238
GRANDE	PEQUENO	0	2523616	0.028459
GRANDE	MEDIO	0	2523616	0.015748
GRANDE	GRANDE	0	2523616	0.011868

O pré-processamento do padrão para obter a tabela de máscara M ocorre na etapa do código. A fase de pesquisa é constituída por um anel em que i varia de 1 até n, com incrementos unitários, no qual cada caractere é analisado e o registrador é alterado para a análise do próximo caractere. Desta forma, temos que o custo do algoritmo é O(n) uma vez que as operações podem ser realizadas em O(1) e levando-se em consideração que o padrão cabe em poucas palavras do computador.

3. Conclusão

Concluímos com o trabalho que os algoritmos Shift-And e Boyer-Moore-Horspool-Sunday são os melhores. Analisando as tabelas apresentadas acima, percebemos que o algoritmo força bruta tem os piores resultados, o algoritmo Boyer-Moore-Horspool-Sunday possui os melhores resultados, o que faria dele a escolha do grupo e o Shift-And seria uma boa escolha caso o objetivo fosse que as comparações fossem similares para diferentes tamanhos de padrão.

Uma das dificuldades foram os problemas ao tentar executar o código no windows, ele compila mas não executa, visto que ocorre um erro e o mesmo não se repete ao executar no linux. Mas a maior dificuldade do grupo foi o Shift-And, já que dos quatro algoritmos foi o que mais apresentou erros, provavelmente por ter uma implementação diferente da que estamos acostumados, utilizando operadores >> << e etc. Mas no geral, o trabalho foi simples de implementar e testar.