Universidade De São Paulo – USP
Instituto de Ciências Matemáticas e da Computação ICMC
Programa de Pós-Graduação

# Resultados encontrados pelos algoritmos Horspool e Boyer-Moore para encontro de substrings em um texto

## SCC 5900 - Projeto de Algoritmos

Matheus Ricardo Uihara Zingarelli - 5377855
Raíza Tamae Sarkis Hanada - 7493723

São Carlos, SP
16 de maio de 2010

# 1.Introdução

O presente trabalho possui o objetivo de comparar os resultados encontrados para os algoritmos Horspool e Boyer-Moore, implementados na Parte 1 do trabalho. Ambos os algoritmos possuem a finalidade de encontrar uma determinada palavra em um texto.

Neste trabalho, são consideradas válidas apenas palavras que contém letras de A a Z (maiúsculas e minúsculas), hífen, e os dígitos de 0 a 9 e tamanho maior ou igual a dois caracteres. São analisadas as velocidades que cada algoritmo executa a busca de acordo com o tamanho das palavras e a frequência delas no texto (cem palavras mais frequente e cem menos frequentes).

Os gráficos para cada experimento A, B e C realizado são apresentados na seção 2 e os resultados são discutidos na seção 3.

# 2.Gráficos Solicitados

Seguem abaixo os gráficos solicitados. Os dados utilizados para formar os gráficos estão disponíveis nos textos parte2.txt, parte2b.txt e parte2c.txt e nas planilhas partea.xlsx, parteb.xlsx e partec.xlsx. Observe que para os experimentos B e C estamos enviando 2 gráficos. A justificativa para isto foi devido a alguns dados isolados apresentarem um comportamento fora do normal da maioria dos gráficos. Como os testes foram realizado em uma máquina virtual que possuía Linux, portanto, com capacidade limitada, pode ser que isto tenha influenciado nos testes. Portanto, para estes experimentos temos um gráfico contendo todos os dados e outro contendo os dados que não extrapolaram certo limiar.

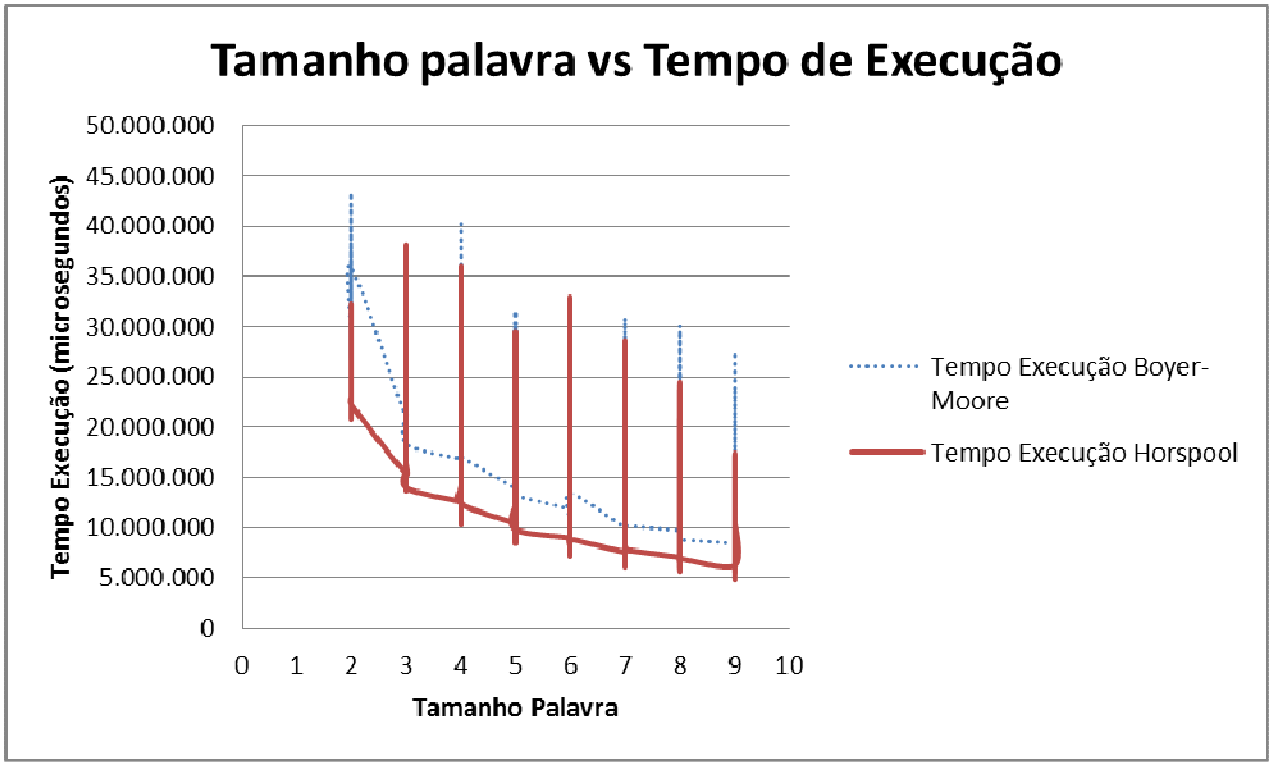


Figura 1 - Gráfico de tempo de execução de busca de palavras de acordo com o seu tamanho.

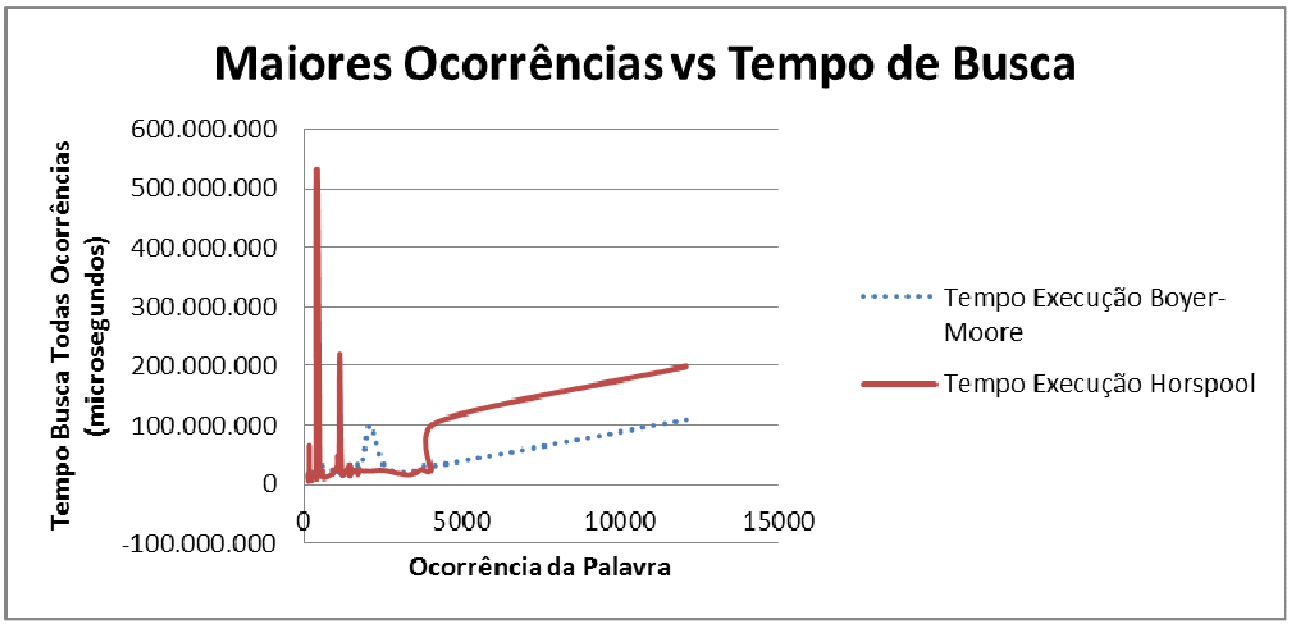


**Figura 2 - Gráfico de tempo de execução de busca de palavras de acordo com a frequência de ocorrência. Resultados para as 100 palavras mais frequentes.**

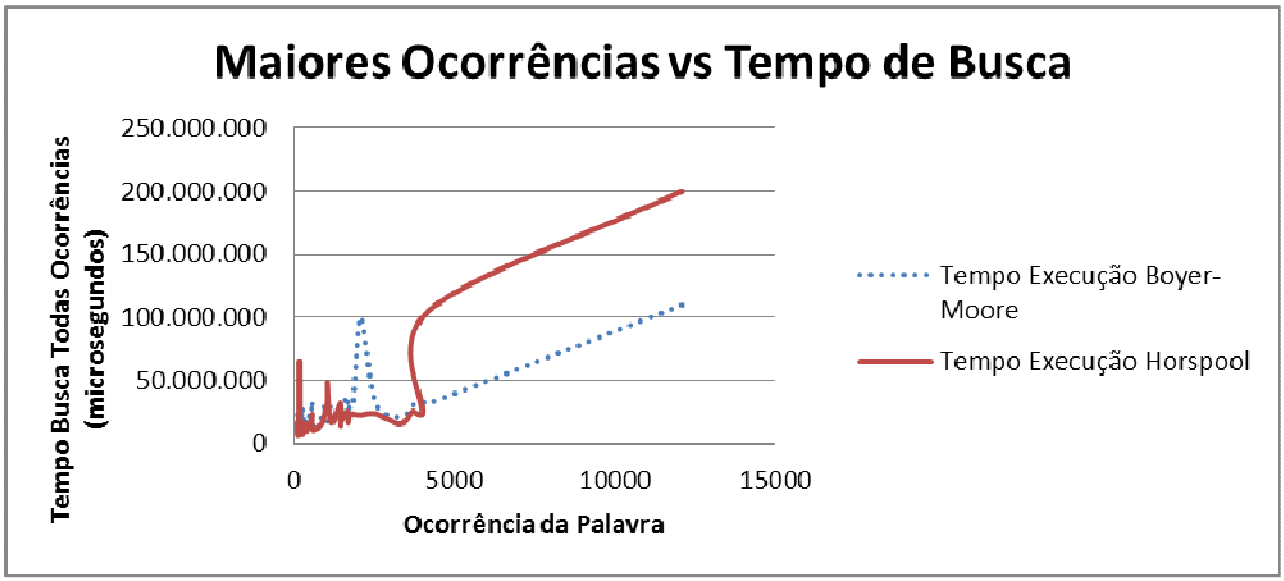


**Figura 3 - Gráfico de tempo de execução de busca de palavras de acordo com a frequência de ocorrência. Resultados para as 100 palavras mais frequentes. Dados que extrapolaram 200.000.000 microssegundos foram eliminados.**

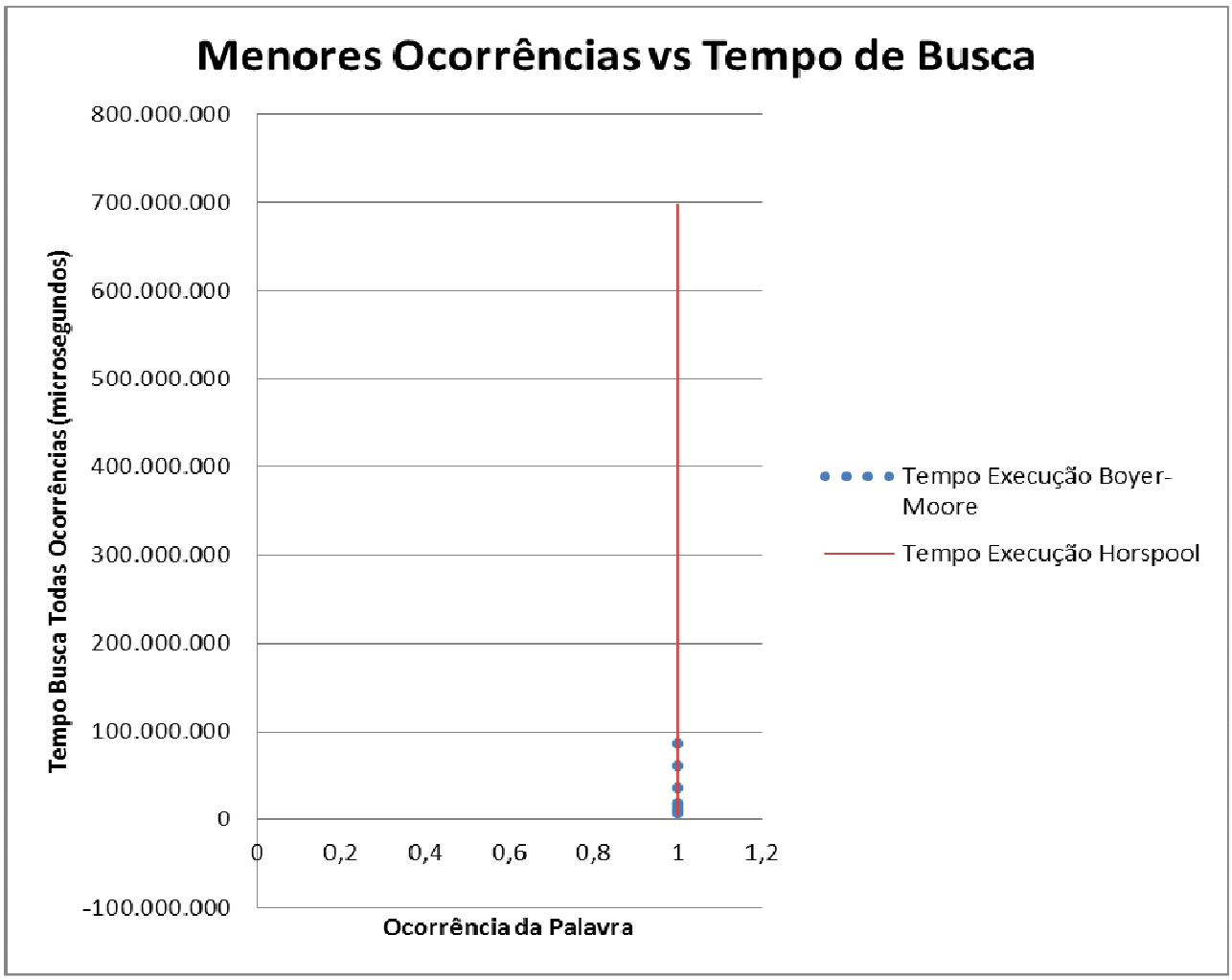


**Figura 4 - Gráfico de tempo de execução de busca de palavras de acordo com a frequência de ocorrência. Resultados para as 100 palavras menos frequentes.**

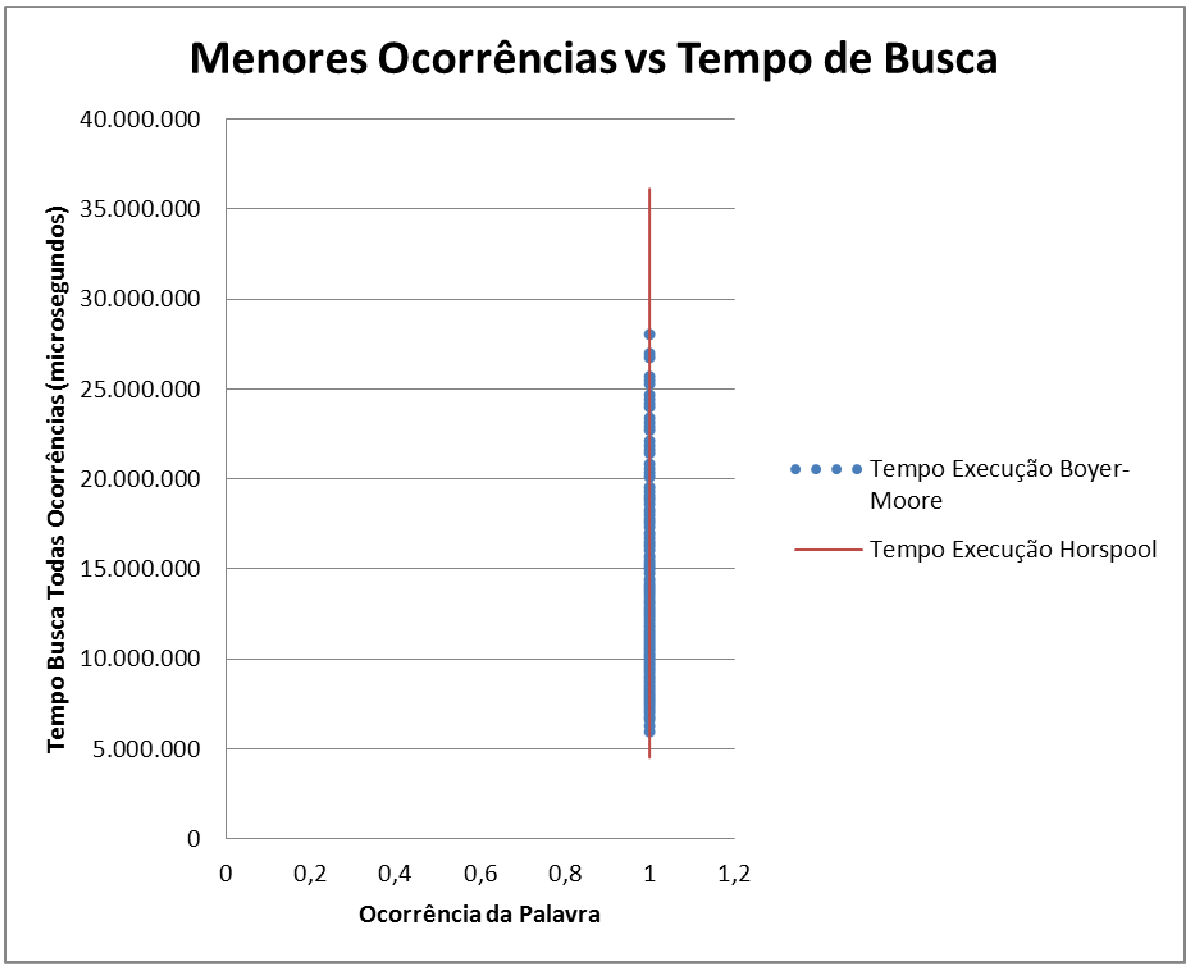


**Figura 5 - Gráfico de tempo de execução de busca de palavras de acordo com a frequência de ocorrência. Resultados para as 100 palavras menos frequentes. Dados que extrapolaram 40.000.000 microssegundos foram eliminados.**

## 3. Discussão dos Resultados

*Experimento A – Tempo de execução em relação ao tamanho da palavra*

Podemos notar que é formada uma curva decrescente na Figura 1, ou seja, para palavras pequenas, o tempo de execução para encontrar todas as suas ocorrências é bem maior do que para palavras grandes. Isto é factível, pois para palavras pequenas o número de comparações que ambos os algoritmos têm que fazer é maior, bem como o tamanho do shift que podem dar. Analisando a performance dos dois algoritmos implementados, vemos que ambos se equilibraram na maioria dos casos, com o de Boyer-Moore se apresentando um pouco mais demorado em alguns dados.

*Experimento B - Tempo de execução das 100 palavras mais frequentes*

A Figura 2 mostra o gráfico que contém as 100 palavras que mais ocorreram no texto, sendo que a palavra de maior ocorrência aparece 12090 vezes e a de menor ocorrência, 127 vezes. Intuitivamente, é para se pensar que quanto mais ocorrências são encontradas, mais tempo o programa gasta para alocar memória e armazenar esta ocorrência na lista de ocorrências. É o que se observa em partes no gráfico, ao se notar o aspecto de curva crescente que ele faz. Das 100 palavras, as que ocorreram com menos frequência levaram menos tempo para serem

processadas na maior parte dos casos. Observam-se dois aspectos interessantes, até 2000 ocorrências de uma palavra, o tempo de execução apresenta uma série de variações. Já de 2000 ocorrências em diante, o gráfico aparenta estabilidade, saltando novamente a partir de ocorrências maiores do que 4000, quando seu crescimento se mantém uniforme. Até 4000 ocorrências, os algoritmos se mostraram equilibrados, porém, após 4000, o algoritmo de Horspool mostrou-se mais ineficiente.

Como se pode observar na Figura 2, o gráfico apresenta dois picos nas palavras de menor ocorrência, sendo que nestes picos, a diferença entre Horspool e Boyer-Moore é muito alta. Como os algoritmos foram executados em uma máquina virtual rodando Linux, com muitas limitações de hardware, entendemos que estes picos devem ser ocorrências localizadas, devido a alguma limitação do hardware. Portanto, criamos um novo gráfico eliminando as ocorrências que ultrapassassem o limiar de 200.000.000 microssegundos (os dois primeiros picos vistos no gráfico da Figura 2). O resultado pode ser observado na Figura 3.

*Experimento C – Tempo de execução das 100 palavras menos frequentes*

Infelizmente o gráfico da Figura 4 dá pouca base para análise. As 100 palavras que ocorreram com menos frequência no texto, ocorreram apenas 1 vez. Novamente, alguns dados possuíam valores muito elevados, com uma diferença entre Horspool e Boyer-Moore muito alta. Com isso, criamos um novo gráfico na Figura 5, em que eliminamos as ocorrências que ultrapassassem 40.000.000 microssegundos. Foram eliminadas 5 ocorrências. Pode-se observar na Figura 5 que o tempo de execução possui um grande intervalo de variação, de 4.500.000 microssegundos a 36.000.000 microssegundos. Provavelmente, as ocorrências que levaram menos tempo de execução devem ser aquelas que aparecem ao final do texto, já que necessitam de menos comparações a ser feita. O algoritmo de Boyer-Moore se mostrou mais eficiente.