Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação
SCC0224 - Estruturas de Dados II

Relatório Projeto 2

Alunos:
Bruno Ideriha Sugahara nº 10310759
Raul Ribeiro Teles nº 13688176
Vinicius de Moraes nº 13749910

Professor: Robson L. F. Cordeiro

Junho
2023

Conteúdo

1 Introdução

O relatório está dividido em duas partes, sendo que a primeira contém 4 implementações de um algoritmo de busca sequencial, enquanto a segunda parte contém 3 implementações de um algoritmo de busca por hashing, todos implementados na linguagem c e em arquivos diferentes.

As implementações foram realizadas a partir dos templates disponíveis, os quais incluem funções para contagem de tempo e leitura de arquivos. Os arquivos utilizados também foram somente os disponibilizados, que incluem arquivos de entrada e de saída com inteiros e strings.

O código está organizado em um arquivo makefile, no qual pode ser usado o comando "make"e o exercício correspondente (make 1a, make 1c, make 2b...) para compilar e executar o programa.

A criação dos gráficos e tabelas foi realizadas através do google sheets conforme as especificações do projeto.

2 Parte I - Análise de Algoritmos de Busca Sequencial

Foram realizadas as seguintes implementações:

1.a - Busca sequencial simples
1.b - Busca sequencial com realocação por meio do método mover-para-frente
1.c - Busca sequencial com realocação por meio do método da transposição
1.d - Busca sequencial utilizando índice primário

As implementações foram relativamente simples e não houve maiores dificuldades.

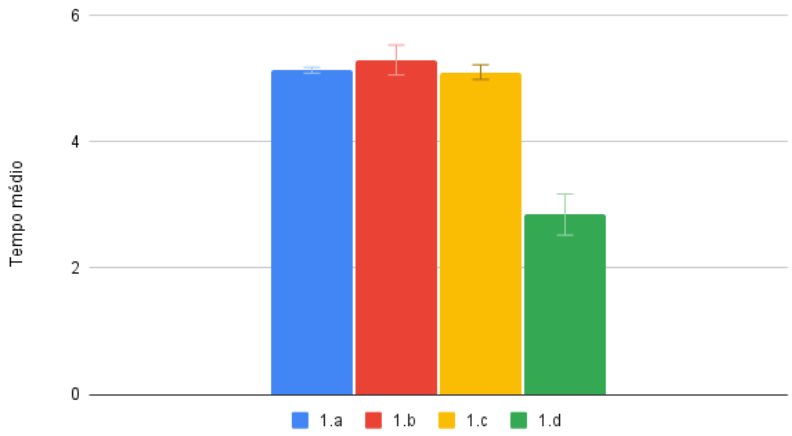
2.1 Tabela de medição

Para cada implementação foram realizadas, no mesmo dispositivo, 3 medições do tempo de busca em segundos e calculados a média e o desvio padrão. Os resultados foram organizados na seguinte tabela:

	Tempo de busca (s)			
	1.a	1.b	1.c	1.d
	5,0985	5,0913	5,3029	3,1950
	5,1320	5,5620	5,1008	2,8467
	5,1893	5,2938	5,1021	2,5437
Tempo médio	**5,1320**	**5,2938**	**5,1021**	**2,8467**
Desvio padrão	**0,0459**	**0,2361**	**0,1163**	**0,3259**

2.2 Gráfico de barras

Tempo médio de busca por implementação:



2.3 Análise

É possível observar que a implementação 1.d que utiliza índice primário na busca foi a mais rápida, apresentando um tempo de busca médio quase duas vezes menor do que as outras, apesar de um desvio padrão um pouco maior o que pode indicar maior inconsistência. As implementações 1.a, 1.b e 1.c apresentaram desempenho muito semelhantes.

3 Parte II - Análise de Algoritmos de Busca com Espalhamento

Foram realizadas as seguintes implementações:

2.a - Busca com espalhamento estático fechado com overflow progressivo (hash por divisão e multiplicação)
2.b - Busca com espalhamento estático fechado com hash duplo
2.c - Busca com espalhamento estático aberto com encadeamento em lista linear não ordenada (hash por divisão e multiplicação)

Observações sobre as implementações:

2a: A maior dificuldade foi testar os resultados devido ao volume de dados.

2b: A implementação foi análoga à anterior no entanto a função hash em

si ficou mais complexa.

3c: Houve certa dificuldade na criação do TAD de lista encadeada. Com essa parte implementada, as função da tabela hash consistem apenas em calcular a posição e chamar a função adequada do TAD de lista encadeada.

3.1 Tabelas de medição

Para cada implementação foram realizadas, no mesmo dispositivo, 3 medições do número de colisões; do tempo de inserção e de busca em segundos; e do número de palavras encontradas. Logo após foram calculados a média e o desvio padrão para cada um dos resultados e organizados nas seguintes tabelas:

	Colisões				
	2.a - divisão	2.a - multiplicação	2.b	2.c - divisão	2.c - multiplicação
	3172570	4911294	1549057	39582	49495
	3172570	4911294	1549057	39582	49495
	3172570	4911294	1549057	39582	49495
Média	**3172570**	**4911294**	**1549057**	**39582**	**49495**
Desvio padrão	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**

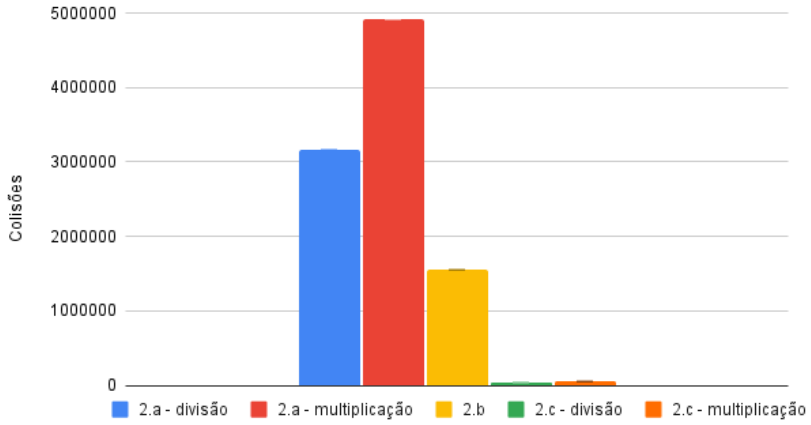
	Tempo de inserção (s)				
	2.a - divisão	2.a - multiplicação	2.b	2.c - divisão	2.c - multiplicação
	0,0238	0,1673	0,0877	0,0058	0,0086
	0,0238	0,1673	0,0867	0,0054	0,0084
	0,0239	0,1688	0,0869	0,0059	0,0090
Média	**0,0238**	**0,1673**	**0,0869**	**0,0058**	**0,0086**
Desvio padrão	**0,0001**	**0,0009**	**0,0005**	**0,0003**	**0,0003**

	Tempo de busca (s)				
	2.a - divisão	2.a - multiplicação	2.b	2.c - divisão	2.c - multiplicação
	0,1520	0,5940	0,2665	0,0611	0,1912
	0,1530	0,5870	0,2686	0,0618	0,1916
	0,1492	0,5946	0,2668	0,0658	0,1883
Média	**0,1520**	**0,5940**	**0,2668**	**0,0618**	**0,1912**
Desvio padrão	**0,0020**	**0,0042**	**0,0011**	**0,0025**	**0,0018**

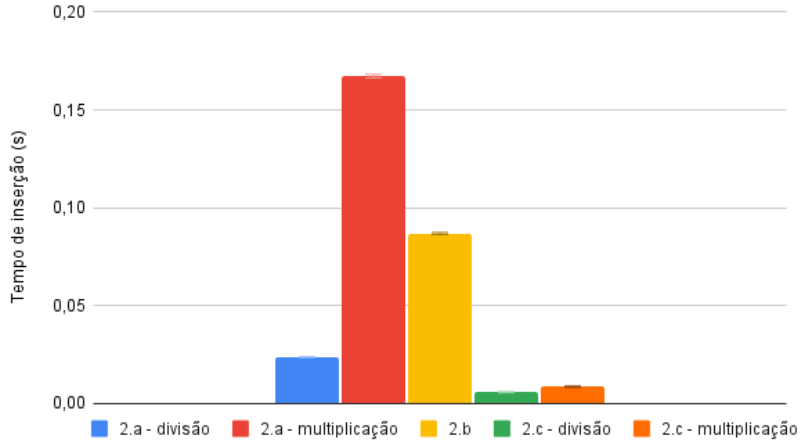
	Palavras encontradas				
	2.a - divisão	**2.a - multiplicação**	**2.b**	**2.c - divisão**	**2.c - multiplicação**
	3753218	3753218	3753218	3753218	3753218
	3753218	3753218	3753218	3753218	3753218
	3753218	3753218	3753218	3753218	3753218
Média	**3753218**	**3753218**	**3753218**	**3753218**	**3753218**
Desvio padrão	**0,0000**	**0,0000**	**0,0000**	**0,0000**	**0,0000**

3.2 Gráficos de barras

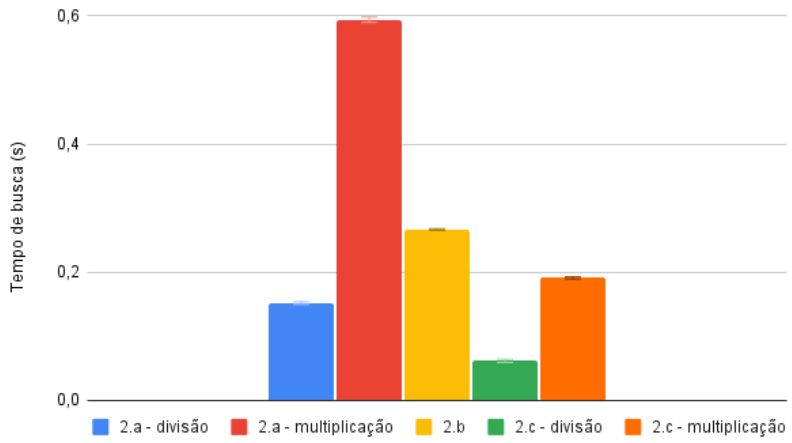
Número de colisões por implementação:



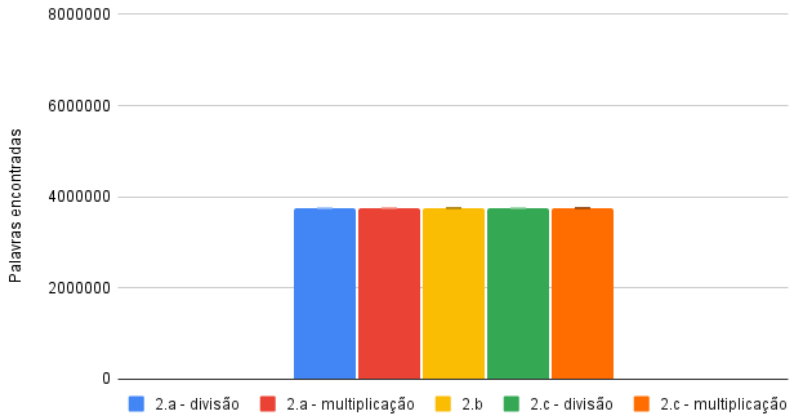
Tempo de inserção por implementação:



Tempo de busca por implementação:



Número de palavras encontradas por implementação:



3.3 Análise

O número de colisões foi muito mais alto para ambas as implementações de 2.a e para 2.b, sendo relativamente insignificante nas versões da 2.c.

Em relação aos tempos de inserção e implementação, a 2.a - multiplicação foi disparadamente a versão que demorou mais tempo, seguida da 2.b. A 2.a - divisão apresentou tempos menores, mas consideravelmente maiores do que os da 2.c - divisão. A implementação 2.c - multiplicação se mostrou uma das mais rápidas para inserção, porém seu tempo de busca é mediano.

Em relação ao número de palavras encontradas, todas obtiveram o mesmo desempenho, encontrando o mesmo número de palavras.

No geral, todos os resultados apresentaram desvio padrão baixo, indicando certa consistência mesmo com um pequeno número de medições.

4 Conclusão

Após considerar os resultados da busca sequencial, é possível notar que apesar da implementação 1.d se destacar entre as demais, todas elas são consideravelmente mais lentas do que os algoritmos de busca com espalhamento.

Em relação às implementações de busca com espalhamento a 2.a - multiplicação demonstra incontestávelmente o pior desempenho, enquanto que as implementações 2.c, principalmente com hash por divisão apresentaram os melhores resultados.