UNIVERSIDADE FEDERAL DE OURO PRETO
DEPARTAMENTO DE CIÊNCIA DE COMPUTAÇÃO
ESTRUTURA DE DADOS II

TRABALHO PRÁTICO 1:

Pesquisa Externa

Alunos:

Brenda Sotero Ferreira,

Gabriel Ferreira Pereira,

Gustavo Vieira Nascimento,

Pedro Henrique Oliveira da Silva,

Stanley Silva Sampaio.

Ouro Preto, MG

04 de novembro de 2021

1 Introdução

Este relatório tem por objetivo descrever o que foi realizado na segunda fase do trabalho prático 1, da disciplina Estrutura de Dados II. O trabalho em si consiste em suas etapas:

- Fase 1: implementação dos métodos de Pesquisa Externa: Acesso Sequencial Indexado, Árvore Binária de Pesquisa Externa, Árvore B e Árvore B*;
- Fase 2: realizar uma análise experimental da complexidade de desempenho desses métodos.

Para que essa análise seja feita adequadamente é importante que alguns requisitos do trabalho sejam atendidos. Tais requisitos dizem respeito aos arquivos e seus tamanhos, um arquivo deve conter 100 registros, outro 1000, 10.000, 100.000 e 1.000.000. Deve-se considerar também três formas de ordenação de arquivos (quando necessário), de acordo com suas chaves de pesquisa: ordenação crescente, decrescente e aleatório.

A partir daí, testes serão realizados considerando cada conjunto de informações. Como? É bem simples: para cada conjunto de dados, ou seja, para um arquivo com 100 registros, por exemplo, que esteja ordenado de forma crescente, ocorrerá uma pesquisa automática de 10 chaves que estão contidas no arquivo. Essas chaves devem ser bem diferentes e estarem espalhadas pelo arquivo. Após feita a pesquisa, a ideia é obter a média para esse conjunto de dados. E seguindo assim para os demais conjuntos.

Os parâmetros para se obter as médias são:

- quantidade de transferências de ítens da memória externa para a interna;
- quantidade de comparações entre chaves de pesquisa;
- tempo de execução.

Estes parâmetros devem ser analisados tanto para a criação de índices quanto para a pesquisa em si.

Cada pesquisa deve retornar se a chave foi encontrada ou não e, caso tenha sido achada, deve imprimir o registro completo da chave de pesquisa e as informações citadas acima. Lembrando que é tanto para índice quanto para a pesquisa. A linha de comando que deve ser usada para fazer a

pesquisa também deve ser seguida (Fig.1). Onde o método é o *método* em questão a ser utilizado, *quantidade* é o tamanho de registro do arquivo, *situação* a ordenação que se encontra o arquivo e *-P* argumento opcional para que a chave seja representada na tela.

pesquisa <método> <quantidade> <situação> <chave> [-P]

Este trabalho está dividido da seguinte forma:

- Introdução: que contém o problema e as especificações para a análise desejada;
- Desenvolvimento: com informações das chaves de pesquisas usadas e o cálculo da média para cada conjunto de dados por método;
- Conclusão: comparando os resultados obtidos acima e apontando quais as maiores dificuldades ao desenvolver o experimento.

2 Desenvolvimento

Como dito anteriormente, a primeira parte deste trabalho consiste na implementação dos métodos de Pesquisa Externa estudados até aqui. Já nesta parte de desenvolvimento da segunda etapa do trabalho, será apresentado os resultados dos testes feitos para cada um dos métodos, considerando os requisitos mostrados no tópico anterior.

As chaves de pesquisa foram definidas considerando as ordens crescente, decrescente e aleatório e os arquivos de 100, 1.000, 10.000, 100.000 e 1.000.000. As chaves foram escolhidas de maneira que a sua distribuição pelo arquivo seja uniforme, garantindo que não há muitos registros no começo ou fim do arquivo. E estão definidas nas imagens Fig.1 e Fig.2. Os testes referentes aos arquivos de 1 milhão de registros não foram realizados devido à limitação de memória e poder computacional para poder simular os métodos de pesquisa.

		100			1000			10000
Crescente	Decrescente	Aleatório	Crescente	Decrescente	Aleatório	Crescente	Decrescente	Aleatório
3	999997	36440	55	999945	128693	540	999460	830858
18	999982	89173	101	999899	187740	1402	998598	236721
25	999975	611583	265	999735	370536	1579	998421	389585
47	999953	594325	345	999655	54082	2950	997050	106688
55	999945	128693	495	999505	390434	3577	996423	635196
64	999936	8640	548	999452	65123	5821	994179	240550
78	999922	413185	678	999322	193572	6661	993339	217973
86	999914	705404	741	999259	414640	7124	992876	34124
94	999906	947740	851	999149	322083	8872	991128	973192
97	999903	698587	923	999077	177755	9210	990790	614711

Figura 1: Chaves de 100 a 10.000

		100000			1000000
Crescente	Decrescente	Aleatório	Crescente	Decrescente	Aleatório
5640	994360	617597	17650	982350	40615
11458	988542	108428	182550	817450	378564
23546	976454	842761	221440	778560	265592
35412	964588	296935	339086	660914	762554
45789	954211	533603	511841	488159	750084
54123	945877	290074	597810	402190	702485
68795	931205	122472	704325	295675	943894
74255	925745	239392	890901	109099	287224
84561	915439	146650	903550	96450	823572
96541	903459	870871	971005	28995	456253

Figura 2: Chaves de 100.000 e 1.000.000

Como não foi possível rodar os testes no arquivo com um milhão de registros, eles não serão considerados para as análises experimental e comparativa. Da mesma maneira a árvore B* que apresentou erros nos testes realizados.

A seguir serão listadas as tabelas com as informações adquiridas em cada método.

2.1 Acesso Sequencial Indexado

Para um arquivo ser pesquisado utilizando o método de Acesso sequencial indexado, ele deve estar ordenado de maneira crescente, portanto não é possível fazer os testes com arquivos decrescentes e aleatórios.

2.1.1 Índice

Acesso sequencial Indexado		Média
Pesquisa Nº	Variável	cresc.
100	Transferências no Índice	100
	Comparações no Índice	100
	Tempo de execução no Índice	0,000499
1000	Transferências no Índice	1000
	Comparações no Índice	1000
	Tempo de execução no Índice	0,001946
10000	Transferências no Índice	10000
	Comparações no Índice	10000
	Tempo de execução no Índice	0,049453
100000	Transferências no Índice	100000
	Comparações no Índice	100000
	Tempo de execução no Índice	0,160179
1000000	Transferências no Índice	0
	Comparações no Índice	0
	Tempo de execução no Índice	0

- Transferências: As transferências são constantes, sempre do tamanho do arquivo
- Comparações: As comparações também são constantes, sendo realizadas no momento de criar as páginas
- Tempo de Execução: Apesar do tempo aumentar, a proporção com que isso acontece é menor do que o aumento no número de registros, com exceção da alteração de 1000 para 10000 registros, que registrou um aumento de 25x

2.1.2 Pesquisa

Acesso sequencial indexado		Média
Pesquisa Nº	Variável	cresc.
100	Transferências na Pesquisa	1
	Comparações na Pesquisa	18,9
	Tempo de execução na Pesquisa	0,0000596
1000	Transferências na Pesquisa	1
	Comparações na Pesquisa	129,7
	Tempo de execução na Pesquisa	0,000035
10000	Transferências na Pesquisa	1
	Comparações na Pesquisa	1198,2
	Tempo de execução na Pesquisa	0,0000411
100000	Transferências na Pesquisa	1
	Comparações na Pesquisa	12507,8
	Tempo de execução na Pesquisa	0,0000752
1000000	Transferências na Pesquisa	0
	Comparações na Pesquisa	0
	Tempo de execução na Pesquisa	0

- Transferências: Sempre vai ocorrer somente uma transferência na pesquisa, pois é a leitura de uma única página no arquivo;
- Comparações: As comparações crescem de maneira linear semelhante aos arquivos, por volta de 10x toda vez que os arquivos aumentam em 10x;
- Tempo de Execução: Apesar das comparações aumentarem linearmente, o tempo de execução não cresce de maneira constante, sempre ficando abaixo de 0.1 milisegundos.

2.2 Arvore Binária de Pesquisa Externa

2.2.1 Índice

Árvore Binária		Média		
Pesquisa Nº	Variável	cresc.	decresc.	aleat.
100	Transferências no Índice	100	100	100
	Comparações no Índice	10309	15259	2030
	Tempo de execução no Índice	0,0027642	0,003418	0,00107
1000	Transferências no Índice	1000	1000	1000
	Comparações no Índice	1003099	1502599	31666
	Tempo de execução no Índice	0,00272	0,003353	0,001035
10000	Transferências no Índice	10000	10000	10000
	Comparações no Índice	100030999	150025999	426590
	Tempo de execução no Índice	0,00048	0,239581	0,009804
100000	Transferências no Índice	100000	100000	100000
	Comparações no Índice	1410375407	2115358111	5425059
	Tempo de execução no Índice	17,983592	22,933125	0,122655
1000000	Transferências no Índice	0	0	0
	Comparações no Índice	0	0	0
	Tempo de execução no Índice	0	0	0

- Transferências: Da mesma forma que os outros método a transferência vai ser o mesmo que o número de arquivos
- Comparações: O número de comparações cresce com um fator de 100x em todos os casos com exceção da alteração de 10000 para 100000 que segue o crescimento de 10x do tamanho do arquivo
- Tempo de Execução: O tempo de execução não apresentou grandes alterações até o teste com 100000
- Diferenças de ordenação: Em todas as métricas não constantes o teste com o arquivo aleatório apresentou métricas dezenas de vezes melhores.

2.2.2 Pesquisa

Árvore Binária		Média		
Pesquisa Nº	Variável	cresc.	decresc.	aleat.
100	Transferências na Pesquisa	1	1	1
	Comparações na Pesquisa	63,4	65,4	15,2
	Tempo de execução na Pesquisa	0,0000642	0,000074	0,0000712
1000	Transferências na Pesquisa	1	1	1
	Comparações na Pesquisa	505,4	507,4	18,1
	Tempo de execução na Pesquisa	0,0000697	0,0000609	0,0000642
10000	Transferências na Pesquisa	1	1	1
	Comparações na Pesquisa	4780,2	4779,2	20,9
	Tempo de execução na Pesquisa	0,000446	0,0004001	0,0004582
100000	Transferências na Pesquisa	1	1	1
	Comparações na Pesquisa	50018	50019	25,7
	Tempo de execução na Pesquisa	0,0037509	0,0035997	0,0075426
1000000	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	0	0	0
	Tempo de execução na Pesquisa	0	0	0

- Transferências: Sempre vai ocorrer somente uma transferência na pesquisa, pois é a leitura de uma única página no arquivo;
- Comparações: As comparações crescem de maneira linear junto com o tamanho do arquivo, em um fator de 10x, com exceção do arquivo aleatório que apresenta um crescimento apesar de ser pequeno;
- Tempo de Execução: O tempo de execução não apresenta grandes alterações de 100 para 1000, após isso ele cresce linearmente com o tamanho do arquivo.

2.3 Arvore B

A implementação da árvore B não foi feita de forma completa devido ao seu alto nível de complexidade. Ela foi feita levando em conta a localização dos arquivos para a pesquisa que, no caso, já estão na memória interna, não precisando de sistema de paginação ou de acesso à memória externa. E por esse motivo as transferências de pesquisa serão sempre iguais a zero.

2.3.1 Índice

Árvore B		Média		
Pesquisa Nº	Variável	cresc.	decresc.	aleat.
100	Transferências no Índice	100	100	100
	Comparações no Índice	2546	2644	2415
	Tempo de execução no Índice	0,0014	0,0013	0,0014171
1000	Transferências no Índice	1000	1000	1000
	Comparações no Índice	38084	35468	33144
	Tempo de execução no Índice	0,0145	0,0163	0,0075915
10000	Transferências no Índice	10000	10000	10000
	Comparações no Índice	505854	438584	427444
	Tempo de execução no Índice	0,1551	0,1571	0,1599469
100000	Transferências no Índice	100000	100000	100000
	Comparações no Índice	6314513	5213591	5190390
	Tempo de execução no Índice	1,5235	1,5765	1,0165731
1000000	Transferências no Índice	0	0	0
	Comparações no Índice	0	0	0
	Tempo de execução no Índice	0	0	0

- Transferências: A transferência é igual ao número de arquivos já que não esta sendo utilizado o sistema de paginação;
- Comparações: As comparações crescem em um rítmo similar ao tamanho do arquivo;
- Tempo de Execução: O mesmo é válido para o tempo de execução, cresce sempre em fatores de aproximadamente 10x.

2.3.2 Pesquisa

Árvore B		Média		
Pesquisa Nº	Variável	cresc.	decresc.	aleat.
100	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	12,7	12,9	12,2
	Tempo de execução na Pesquisa	0,0002	0,0006	0,0000183
1000	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	19,2	20,1	18,8
	Tempo de execução na Pesquisa	0,0006	0,0003	0,0000309
10000	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	27,4	28,2	26,8
	Tempo de execução na Pesquisa	0,0003	0,0003	0,0000337
100000	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	35,3	33,7	33,4
	Tempo de execução na Pesquisa	0,0002	0,0002	0,0000353
1000000	Transferências na Pesquisa	0	0	0
	Comparações na Pesquisa	0	0	0
	Tempo de execução na Pesquisa	0	0	0

- Transferências: Independente do tamanho do arquivo ou de sua ordenação, seu valor sempre será zero como dito acima;
- Comparações: A média de comparações considerando o tamanho dos arquivos tem uma variância pequena, ou seja, não cresce muito de acordo com o crescimento da quantidade de registros. Isso também pode ser visto nas ordenações, com pouca variação. Em todos os casos, quando o arquivo se encontra na ordem aleatória, ele tem um menor número de comparações;
- Tempo de Execução: Também tem uma variação pequena considerando o tamanho e ordem dos arquivos. Os menores tempos estão na coluna da ordem aleatória.

2.4 Arvore B*

Durante o desenvolvimento do trabalho foram encontrados diversos problemas na implementação do algoritmo de Árvore B* referentes à construção da árvore. Devido a estes problemas não

foi possível executar os testes para avaliar a performance do método de busca.

3 Conclusão

De acordo com as análises do capítulo de Desenvolvimento, pode-se fazer uma comparação entre os métodos e determinar em quais situações cada um deles é melhor utilizado. Como métrica de comparação, foi utilizado o número de comparações de pesquisa e criação de índices, já que o tempo de execução é uma métrica muito volátil que pode mudar a cada vez que o programa é executado. Com isso é possível classificar o melhor método dependendo da natureza da aplicação. Caso sejam necessárias muitas construções de índices, o método que apresenta o melhor desempenho é o de Acesso sequencial indexado, já que os métodos de Árvore são muito custosos nesta etapa. No entanto, caso sejam efetuadas muitas pesquisas os métodos de Árvore são uma opção melhor, sendo que a Árvore B aparenta ser uma implementação mais adequada, com menos comparações tanto na criação dos índices quanto na pesquisa.

Alguns problemas foram encontrados no decorrer deste trabalho, como:

- Arquivo com 1.000.000 de registros: ao realizar os testes neste arquivo específico, ele ficou executando por um tempo indeterminado. E por esse motivo não foi possível finalizar os testes nele;
- Implementação da árvore B*: problemas na construção da árvore impediu que avançasse para a fase de testes nesse método;
- Organização dos códigos: como o trabalho foi feito de forma remota pelos integrantes do grupo, num certo ponto houve uma inconsistência das versões de código já existentes. Que foi resolvido ao criar um repositório no GitHub.

Fora isso, todas as implementações ocorreram de acordo com as especificações dos métodos e seus códigos dados em sala de aula. Seguindo o que foi aprendido até o momento.