



Universidade Federal do Ceará — Campus Quixadá
Estrutura de Dados Avançados — 2017.1
Projeto EDA, Exercício 2

Bárbara Stéph. Neves, Joyce N. Araújo
Matrículas — 388713 e 383868
26 de Março de 2017

Tema: Resultados Computacionais das Listas de Prioridades

1. Configurações da Máquina

O computador usado para fazer a comparação computacional possui 3,8 GiB de memória, um processador Intel® Core™ i3-4005U CPU @ 1.70GHz × 4, o sistema operacional é o Ubuntu 14.04 LTS, e o tipo de sistema é de 64-bit.

2. Resultados

A seguir, são apresentadas tabelas contendo os tempos computacionais de uma dada estrutura de dados. As implementações feitas utilizaram Vetor Não Ordenado, Vetor Ordenado e Heap Máximo, e todas elas simulam uma aplicação que utiliza Lista de Prioridades.

Foram utilizados 6 arquivos com variados tamanhos para os testes. Assim, as tabelas abaixo especificam o tipo de arquivo em que as estruturas trabalharam, o tipo de operação que realizaram — se foi de inserção (I), alteração (A), remoção (R), e seleção (S) —, e o tempo total que cada estrutura levou para rodar essas listas de prioridades, tempo este dado em milissegundos.

ARQUIVOS DE TAMANHO 100					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	2	0	1	1	4
Vetor Ordenado	3	1	1	0	5
Heap Máximo	2	1	1	1	5

- **Arquivos de tamanho 100:**

Esta tabela indica que os arquivos de tamanho 100 rodou mais rápido com o Vetor Não Ordenado do que com as outras duas estruturas. Na remoção, todas elas possuem tempos iguais (1 milissegundo); na alteração, o Vetor Não Ordenado demora menos; na seleção, quem sai ganhando é o Vetor Ordenado; e, na inserção, Heap e Vetor Não Ordenado empatam.

ARQUIVOS DE TAMANHO 10000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	21	90	122	135	368
Vetor Ordenado	179	152	10	11	352
Heap Máximo	19	62	11	9	101

- **Arquivos de tamanho 10000:**

Aqui claramente fica explícito que o Heap Máximo foi bem mais rápido, tanto no tempo total em que trabalhou nos arquivos de tamanho 10000, quanto em todas as operações feitas. Agora, analisando os dados, o Vetor Não Ordenado possui remoção e seleção bem

lentas (122 e 135 milissegundos), enquanto que o Vetor Ordenado tem a inserção e alteração mais lentas (179 e 152 milissegundos). Já o Heap demora mais na alteração, e é um pouco mais lento que o Vetor Ordenado na remoção.

ARQUIVOS DE TAMANHO 50000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	27	973	1563	3606	6169
Vetor Ordenado	2583	2772	28	26	5409
Heap Máximo	39	990	29	15	1073

- **Arquivos de tamanho 50000:**

Heap Máximo novamente se sai melhor, mas agora com os arquivos de tamanho 50000. Como na tabela anterior, Vetor Não Ordenado foi mais lento removendo e selecionando, enquanto que o Ordenado demorou inserindo e alterando. E, novamente, Heap Máximo demorou na alteração e foi mais lento que o Vetor Ordenado na remoção.

ARQUIVOS DE TAMANHO 100000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	9	3685	5675	14431	23800
Vetor Ordenado	10787	12271	21	25	23104
Heap Máximo	12	3960	31	12	4015

- **Arquivos de tamanho 100000:**

Nessa tabela Heap Máximo foi o mais rápido, sendo que a diferença entre o Vetor Não Ordenado e o Ordenado é bem pequena, como também ficou explícita nas tabelas anteriores. O mesmo padrão se segue para o Vetor Ordenado, que é mais lento na inserção e alteração, mas, dessa vez, Heap Máximo foi mais lento que o Vetor Não Ordenado nas operações de inserção de tamanho 100000.

ARQUIVOS DE TAMANHO 500000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	28	103181	145111	408471	656791
Vetor Ordenado	284813	307691	113	118	592735
Heap Máximo	46	101717	178	35	101979

- Arquivos de tamanho 500000:

Heap Máximo novamente foi o mais rápido dentre as estruturas. Mas na inserção, o Vetor Não Ordenado foi mais rápido que o Heap, e o Ordenado acabou sendo bem lento (aproximadamente 5 minutos), e, na alteração, a diferença entre o Vetor Não Ordenado e o Heap é de apenas 1,46 segundos. O padrão criado nas tabelas anteriores continua o mesmo para o Vetor Ordenado.

ARQUIVOS DE TAMANHO 800000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Vetor NOrdenado	45	308934	390992	956984	1656955
Vetor Ordenado	807543	864855	208	205	1672811
Heap Máximo	77	254372	315	102	254866

- Arquivos de tamanho 800000:

Convertendo os valores de milissegundos para minutos do tempo total que cada estrutura levou para rodar os arquivos com 800000 operações, o Vetor Não Ordenado e Ordenado levaram aproximadamente 28 minutos, e o Heap levou em torno de 4 minutos para executar as operações. Como na tabela anterior, o padrão continua se repetindo para certas operações, tornando o Heap bem mais lento em algumas delas mas, se torna claro que, sem dúvida ele é bem rápido se levar em conta o tempo de execução total.

3. Conclusões

	OperaçõesI	OperaçõesA	OperaçõesR	OperaçõesS
Vetor NOrd.	132	416863	543464	1383628
Vetor Ordenado	1105908	1187742	381	385
Heap Máximo	195	361102	565	174

A tabela acima possui o tempo total que cada operação leva para executar em cada estrutura. A partir da tabela que possui os arquivos de tamanho 10000, foi-se estabelecido um padrão para o Vetor Não Ordenado e o Ordenado. Este padrão especifica que o Vetor Não Ordenado é mais lento nas operações de remoção e seleção, enquanto que o Ordenado é mais lento nas de inserção e alteração.

Pelo menos para o Vetor Ordenado, o padrão não foi alterado nas seguintes tabelas. Mas, tanto para o Heap quanto para o Não Ordenado, começaram a aparecer algumas diferenças. Como mostrado na tabela que possui os tempos totais das operações, Heap Máximo é um pouco mais lento na operação de inserção que o Vetor Não Ordenado, como também é mais lento que o Vetor Ordenado na operação de remoção.

TEMPO TOTAL	
Vetor Não Ordenado	39 minutos
Vetor Ordenado	38 minutos
Heap Máximo	6 minutos

Assim, desde a tabela de tamanho 10000 é que pode perceber essa sutil diferença e lentidão por conta do Heap Máximo. Mas, apesar das diferenças serem de poucos milissegundos, ele continua sendo o mais rápido que as outras estruturas, e isso se torna claro quando é verificado, em minutos, o tempo total que cada uma delas levou para executar todas as operações, como mostrado na tabela acima.