



Universidade Federal do Ceará — Campus Quixadá
Estrutura de Dados Avançados — 2017.1
Projeto EDA, Exercício 3

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 implementações que utilizam Heap Máximo Binário e Heap Máximo Ternário, 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
Heap Máx. Binário	2	1	1	1	5
Heap Máx. Ternário	2	1	1	1	5

- **Arquivos de tamanho 100:**

Esta tabela indica que para os arquivos de tamanho 100 não houve diferença entre o processamento das duas estruturas. Todas as operações possuem tempos iguais, declarando um certo empate nessa simulação.

ARQUIVOS DE TAMANHO 10000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Heap Máx. Binário	19	62	11	9	101
Heap Máx. Ternário	17	267	9	9	302

- **Arquivos de tamanho 10000:**

Há uma pequena diferença entre as duas estruturas com o processamento dos arquivos de tamanho 10000. Apesar de que em todas as operações, menos da de Alteração, o Heap Ternário é bem mais rápido, Heap Binário no tempo total continua sendo menos lento.

ARQUIVOS DE TAMANHO 50000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Heap Máx. Binário	39	990	29	15	1073
Heap Máx. Ternário	29	5317	18	12	5376

- **Arquivos de tamanho 50000:**

Com a tabela acima, podemos observar que nos arquivos de tamanho 50000, o Heap Binário se sai bem melhor que o Heap Ternário por causa do processamento das operações de alteração, já que é o processo mais demorado realizado, sendo que é mais rápido que o Heap Binário nas operações de Inserção, Remoção e Seleção.

ARQUIVOS DE TAMANHO 100000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Heap Máx. Binário	12	3960	31	12	4015
Heap Máx. Ternário	9	35275	8	10	35302

- **Arquivos de tamanho 100000:**

Heap Binário continua sendo mais rápido que Heap Ternário, já que este último sofre uma pequena transformação com a operação de Alteração. O valor, se comparado ao Heap Binário, custa bem mais que o dobro de tempo que ele leva para executar no todo.

ARQUIVOS DE TAMANHO 500000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Heap Máx. Binário	46	101717	178	35	101979

Heap Máx. Ternário	39	976779	38	147	977003
---------------------------	----	--------	----	-----	---------------

- **Arquivos de tamanho 500000:**

O padrão criado nas tabelas anteriores continua o mesmo para o Heap Máximo Ternário. Somando mais uma pequena diferença na operação de Seleção, pois, pela primeira vez, Heap Ternário levou mais tempo para executá-la.

ARQUIVOS DE TAMANHO 800000					
	Inserção	Alteração	Remoção	Seleção	TOTAL
Heap Máx. Binário	77	254372	315	102	254866
Heap Máx. Ternário	66	2435130	60	44	2435300

- **Arquivos de tamanho 800000:**

Como na tabela anterior, o padrão continua se repetindo para as operações de Inserção e Remoção, sendo que, nesta tabela, a Seleção do Heap Ternário se tornou mais rápida. Mas, com base no tempo total que cada estrutura levou para executar as operações, Heap Binário se torna bem mais rápido.

3. Conclusões

	OperaçõesI	OperaçõesA	OperaçõesR	OperaçõesS
Heap Máx. Binário	195	361102	565	174
Heap Máx. Ternário	162	3452769	134	223

A tabela acima possui o tempo total que cada operação leva para executar em cada estrutura. Do qual podemos observar que há algumas operações que o Heap Binário é bem mais rápido e, em outras, o Heap Ternário supera a outra estrutura. Um exemplo é que o tempo de processamento do Heap Binário para executar as operações de Alterações é 10% do

tempo que o Heap Ternário leva para processar a mesma operação. E, nas operações de Remoção, o Heap Ternário utiliza no máximo 24% do tempo que o Heap Binário utiliza.

TEMPO TOTAL	
Heap Máximo Binário	6 minutos
Heap Máximo Ternário	57 minutos

Com base no tempo total dado em minutos da tabela acima, e tomando como apoio os estudos feitos com relação às duas estruturas, Heap Máximo Binário e Heap Máximo Ternário são equivalentes, ou seja, os dois possuem uma pequena diferença tanto de processamento quanto de forma, já que o Heap Binário trabalha com uma Árvore Binária, enquanto que, no Ternário, cada pai possui três filhos.

O que atrasou bastante o funcionamento da estrutura do Heap Ternário, segundo os dados coletados, ocorreu durante a execução da operação de Alteração. Como o prazo solicitado para entrega da implementação feita em código Java e do documento com a análise dos experimentos foi bem curto, não sabemos se o problema que ocorreu nos testes possa ter vindo de alguns métodos feitos erroneamente (como o subir e descer das prioridades), ou mesmo no que altera essas prioridades. O fato é que, tentamos, aprendemos sobre as estruturas, e aqui está o que o trabalho finalizado.