

Estudo Computacional

Equipe: Natanael e Geraldo

Computador utilizado nos testes

Informações Técnicas da Máquina	
Marca	Dell
Processador	Core i7 5ª Geração 2.40 GHz
Memória RAM	8 Gigabytes
Sistema Operacional	Windows 10 64 bits

1. Lista prioridades com vetor não ordenado.

Operações em milissegundos(ms)						
	Operações _100	Operações _10000	Operações _50000	Operações _100000	Operações _500000	Operações _800000
Inserção	2	133	2.251	12.728	387.584	997.759
Alteração	0	31	529	2.142	66.949	170.996
Seleção	1	85	1.690	8.211	223.159	588.187
Remoção	1	37	779	3.662	96.754	259.200

A tabela acima mostra o tempo gasto em milissegundos por cada operação realizada pela estrutura de dados lista de prioridade com vetor não ordenada. Podemos observar o seu desempenho desde atividades pequenas ate atividade com um número elevado de operações. Os dados mostram que seu melhor desempenho é na operação de ALTERAÇÃO, ficando com o segundo melhor desempenho a operação de REMOÇÃO. Pode-se perceber que essa estrutura é mais adequada para uma aplicação que tenha mais operações de ALTERAÇÕES e REMOÇÕES, caso contrário ela não é a estrutura ideal para uma aplicação que tenha mais operações de INSERÇÃO e SELEÇÃO, pois a aplicação ficaria lenta e essa qualidade não é aceita em grandes aplicações.

2. Lista prioridades com vetor ordenado.

Operações em milissegundos(ms)						
	Operações _100	Operações _10000	Operações _50000	Operações _100000	Operações _500000	Operações _800000
Inserção	2	142	1.433	4.865	46.041	1.240.928
Alteração	1	280	2.265	17.786	246.976	1.710.981
Seleção	0	147	452	5.493	465.564	333.525
Remoção	0	13	1.117	4.723	227.870	223.756

A tabela acima mostra os tempos referentes a estrutura de dados lista de prioridade com vetor ordenado gasta para realizar operações de tamanho pequeno até operações com tamanho mais considerado. Com esses dados é possível identificar de modo geral que a operação com melhor desempenho é a operação de REMOÇÃO, os dados também mostram que a operação com segundo melhor desempenho é de SELEÇÃO.

3. Heap Máximo.

Operações em milissegundos(ms)						
	Operações _100	Operações _10000	Operações _50000	Operações _100000	Operações _500000	Operações _800000
Inserção	1	36	349	1.321	33.463	104.136
Alteração	1	29	511	2.231	69.179	161.230
Seleção	1	18	269	1.137	31.437	77.922
Remoção	0	14	280	1.141	27.040	64.596

De modo geral, o Heap máximo conforme estudado em sala é a mais rápida das três estruturas, entretanto, não sabíamos quão grande seria esta diferença de performance, o que testemunhamos a partir da execução das tarefas, em que houve uma drástica redução de tempo em comparação com a Lista ordenada e não ordenada com grandes quantidades de operações. Destacam-se no Heap os quesitos de REMOÇÃO e SELEÇÃO de dados, que demandam muito menos tempo diante das operações de INSERÇÃO e mais ainda das ALTERAÇÕES.

RESULTADOS

1. A partir da leitura da tabela podemos inferir que o tempo de processamento para 500000 operações, cresce vertiginosamente em comparação com o arquivo anterior de 100000 operações.

2. O tempo de processamento é praticamente o mesmo para até 50000 operações entre a Lista Ordenada e a Não Ordenada, a partir de 100000 operações nota-se a vantagem levada pela Lista Não Ordenada.

3. O Heap consegue ser aproximadamente 4,94 vezes mais rápido do que a lista não ordenada e 8,6 vezes mais que lista ordenada na comparação de tempo total.

4. Classificando as operações, obteve-se como resultado:

Inserção – Heap máximo possui o menor tempo de execução em todas as quantidades de operações.

Alteração – Houve um empate entre Heap máximo e lista não ordenada, com heap sendo mais rápido nas operações de 10000, 50000 e 800000. Já a lista não ordenada foi célere nas operações com 100, 100000 e 500000.

Remoção – Domínio do Heap, porém no arquivo com 10000 operações destaca-se a lista ordenada.

Seleção – Novamente domínio do Heap, onde desta vez a lista ordenada ficou a frente no arquivo operações 100.

5. Concluimos por meio dos testes realizados que o Heap Máximo, dentro das estruturas analisadas exalta-se como a mais eficiente no quesito tempo de execução. Isso denota-se pela ampla vantagem apurada na tabela CLASSIFICAÇÃO DAS OPERAÇÕES, com maior quantitativo de citações no resultado das comparações realizadas por arquivo e tarefa.

6. Características apresentadas por cada estrutura:

Heap Máximo: o tempo para executar as alterações geralmente foi de aprox. 249% maior que o tempo de remoção, 206% que o de seleção e 60% que o de inserção.

Lista Ñ Ordenada: possui maior tempo de execução nas operações em que predominam a inserção de elementos na estrutura. A inserção leva 5,83 mais tempo que as alterações, 3,84 vezes mais tempo que a remoção e 1,69 mais tempo que a seleção.

Lista Ordenada: das estruturas testadas foi a que levou mais tempo para excutar os arquivos a partir de 50000 operações, chegando a levar quase 1h para realizar todas as operações do arquivos com 800000 operações.

GRÁFICOS





