



INSTITUTO FEDERAL DE MINAS GERAIS

Bacharelado em Ciência da Computação

Disciplinas: Programação II e Algoritmos e Estrutura de Dados I

Trabalho Prático 2

Professores: Mário Luiz Rodrigues Oliveira e
Walace de Almeida Rodrigues

Aluno: Guilherme Cardoso Silva 0022645

Formiga-MG
17 de junho de 2016

INTRODUÇÃO

Foi proposto a implementação de 4 métodos de ordenação: Bolha, Inserção, QuickSort e o QuickSort Turbinado.

Ambos apresentam um custo diferenciado, porém cada um tem uma característica que se diferencia dos outros.

O método da Bolha apresentam um custo fixo sendo sempre $O(n^2)$.

O método Inserção apresenta um ótimo custo para vetor que já estão próximos de ordenados, porém apresenta um custo elevado para um vetor em ordem decrescente que pode chegar até um valor $O(n^2)$ em seu pior caso.

O método QuickSort trabalha com a sorte, ele escolhe um Pivo aleatoriamente e classifica o resto do vetor como menor e maior que este valor, então ele vai quebrando o vetor e repetindo o processo. Porém caso o valor sorteado para o pivô seja o maior elemento do vetor o custo deste algoritmo também pode chegar a $O(n^2)$. Então quando o vetor já está quase ordenado com os maiores valores no fim seu custo começa a ser grande.

O método QuickSort Turbinado se trata de uma alteração no QuickSort normal, quando o tamanho do vetor começa a trabalhar com valores pequenos seu custo começa a aumentar, pois a chance de um grande ser sorteado é grande. Então quando o tamanho do vetor atinge tamanho menor ou igual a 40 ele chama o método Inserção que para vetores pequenos e quase ordenados seu custo é baixo.

COMPARAÇÃO

Para medir o custo de cada algoritmo em diversos casos, foram gerados vetores de tamanho 500, 1000 10000, 100000 e 300000 elementos.

Os elementos gerados nestes vetores foram gerados através de uma função randômica.

Os algoritmos foram executados 3 vezes cada para gerar resultados médios destas execuções.

Os resultados podem ser observados nas tabelas a seguir:

Bolha

Quantidade de Elementos	Quantidade de Comparações Médias	Quantidade de Trocas Médias	Tempo Médio de Execução	Desvio-padrão de Comparações	Desvio-padrão de Trocas	Desvio-padrão de Tempo
500	125250.00	58550.33	0.00	0.00	235.23	0.00
1000	500500.00	245164.00	0.02	0.00	470.93	0.00
10000	50005000.00	25019652.00	0.74	0.00	4713.57	0.03
100000	5000050000.00	2508917476.33	72.69	0.00	47140.45	0.18
300000	45000150000.00	22477744720.00	703.72	0.00	141421.36	37.04

Inserção

Quantidade de Elementos	Quantidade Comparações Médias	Quantidade de Trocas Médias	Tempo Médio de Execução	Desvio-padrão de Comparações	Desvio-padrão de Trocas	Desvio-padrão de Tempo
500	58876.00	58884.00	0.00	0.00	0.00	0.00
1000	245824.00	245831.00	0.01	0.00	0.00	0.00
10000	25026305.00	25026319.00	0.30	0.00	0.00	0.00
100000	2508984132.00	2508984143.00	29.78	0.00	0.00	0.24
300000	22477944707.00	22477944720.00	319.12	0.00	0.00	10.66

QuickSort

Quantidade de Elementos	Quantidade Comparações Médias	Quantidade de Trocas Médias	Tempo Médio de Execução	Desvio-padrão de Comparações	Desvio-padrão de Trocas	Desvio-padrão de Tempo
500	5162.33	3552.33	0.00	552.32	651.42	0.00
1000	11295.33	7325.00	0.00	585.80	578.96	0.00
10000	152649.33	88292.67	0.00	2190.26	498.86	0.00
100000	2006116.33	1167651.33	0.03	34860.82	45153.91	0.00
300000	6690674.00	3856754.00	0.11	65596.70	197127.36	0.00

QuickSort Turbinado

Quantidade de Elementos	Quantidade Comparações Médias	Quantidade de Trocas Médias	Tempo Médio de Execução	Desvio-padrão de Comparações	Desvio-padrão de Trocas	Desvio-padrão de Tempo
500	3213.67	1592.33	0.00	846.39	454.57	0.00
1000	7556.33	4065.67	0.00	433.97	232.84	0.00
10000	119074.33	60205.00	0.00	2816.96	5730.92	0.00
100000	1602654.67	841954.33	0.03	26843.70	62539.32	0.00
300000	5661739.67	2906621.33	0.11	149862.60	139628.98	0.00

AValiação

Analisando os resultados é possível observar que para o método da bolha o custo basicamente o mesmo que sua complexidade, pois ele realiza todas as comparações e trocas possíveis entre os valores, tendo um custo fixo em 99% de seus casos.

O método Inserção trabalha em seu caso médio na maioria dos seus testes por se tratar de números randômicos a probabilidade que ele caia no seu pior ou no seu melhor caso é pequeno. Uma boa observação sobre o método é que a quantidade de trocas é quase a mesma da quantidade de comparações, pois o método só

realiza uma troca se a comparação for favorável, caso contrário ela considera como aquela parte ordenada e continua o método.

O método QuickSort apresenta um desempenho excelente comparado aos anteriores, como o pivô do método é gerado de forma randômica entre a mediana de 3 valores a probabilidade de que ele caia no seu pior caso é reduzida para uma porcentagem mínima, fazendo com que o método trabalhe entre seu melhor caso e seu caso médio. Quando as partições começam a se tornar pequenas seu rendimento começa a cair, podendo chegar ao seu pior caso com uma porcentagem maior.

O método QuickSorte Turbinado tem como finalidade utilizar o método Inserção para partições menores que 40 elementos, pois se tratam de vetores com uma possibilidade maior de estarem próximos de sua ordenação, resultando em menos comparações e menos trocas entre os elementos. Ele se torna então mais rápido que o QuickSort normal. Como ambos são rápidos a diferença de tempo entre os dois é pouco visível, mas se analisado as trocas e comparações se torna bem mais visível.

CONCLUSÃO

Analisando cada algoritmo em seus devidos casos é possível observar que cada um atua melhor em um determinado ambiente.

O método da bolha é considerado constante, onde sua complexidade varia muito pouco.

O método inserção se mostra eficiente para casos onde o vetor já está ordenado, ou para vetores pequenos que tem uma alta probabilidade de estar perto de sua ordenação.

O método QuickSort apesar de trabalhar com a sorte, apresenta uma grande probabilidade de quase nunca estar em seu pior caso, estando entre seu melhor caso e seu caso médio.

O método QuickSort Turbinado explora a eficiência do método Inserção para cobrir seu alto custo com vetores pequenos, tornando o algoritmo mais veloz e eficiente.

O intuito do trabalho teve como finalidade a pratica destes algoritmos. Onde que para medir seu custo e seu tempo para vários casos é necessário conhecer afundo o que cada algoritmo deve fazer para saber o que se deve contabilizar.

A parte 1 do trabalho exercitou muito a ideia de listas, ponteiros e como eles podem se relacionar. Aplicando três algoritmos (parte 2) que utilizam estas funções.

Após a construção desta TAD e com a quantidade de testes realizados nela é possível passa-la para frente para que possa ser usada por qualquer pessoa sabendo que não serão encontrados erros que influenciem ao desenvolvedor.

Na parte dois também foi bastante exercitado a ideia de pilhas e filas para utilização de coloração de grafos e buscas por profundidade e largura.

Na minha opinião a busca pela informação para realizar todo o trabalho resultou em um crescimento no meu conhecimento muito grande.