Eficiência do Algoritmo Insertionsort

Caio C. Santos[1], Saulo C. Pereira[1], André L. Vieira[1], Rodrigo J. Santos[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 54740-540 – Recife – PE – Brasil

{ccls,scrps,rjbcs}@cin.ufpe.br

Abstract. *This article aims to make a brief analysis of performance insertionsort algorithm for a specific scenario. The analysis is essentially experimental.*

Resumo. *Este artigo visa fazer uma breve análise de desempenho do algoritmo de ordenação insertionsort para um cenário específico. A análise realizada é de caráter fundamentalmente experimental.*

1. Introdução

No meio tecnológico os custos operacionais com algoritmos são bastante influentes na eficiência do processamento. Com base nisto, foi aplicada a estatística descritiva no algoritmo insertionsort para posterior análise de sua real eficiência no âmbito da computação. O insertionsort é um algoritmo baseado na ordenação por inserção muito eficiente quando aplicado a um pequeno número de elementos.

2. Descrição do experimento

O experimento em análise consistiu em rodar o algoritmo de insertionsort para ordenar 300 vetores de elementos aleatórios e tamanho fixo de 30002 elementos. Em cada execução associada a uma das 300 entradas, estávamos interessados em medir o tempo de execução do algoritmo (tempo até terminar a ordenação).

Dado o caráter "randômico" das entradas o tempo de execução será modelado como uma variável aleatória contínua.

Além disso, acreditamos que essa variável aleatória seja normalmente distribuída. Para verificar esta afirmativa usaremos nossos dados experimentais para montar um histograma.

Sabemos que uma variável aleatória normalmente distribuída obedece a uma função cujo gráfico possui formato de sino. Como nosso conjunto de medidas é discreto, montaremos um histograma (Figura 1) e verificaremos a disposição das colunas para ver se estas de fato aproximam razoavelmente o resultado esperado.

Além disso, uma tabela com as medidas estatísticas (Tabela 1) foi confeccionada para ilustrar os resultados.

2.1. Infraestrutura

Todas as medições foram feitas numa máquina com processador Intel Core i3 M 350 @2.27 GHz, 2GB de memória RAM e cujo sistema operacional é Windows 7 Home 32 bits.

2.2. Dados

Os dados usados no experimento assim como os valores da tabela foram disponibilizados no link: http://cin.ufpe.br/~scrps/Estatistica

3. Figuras e Legendas

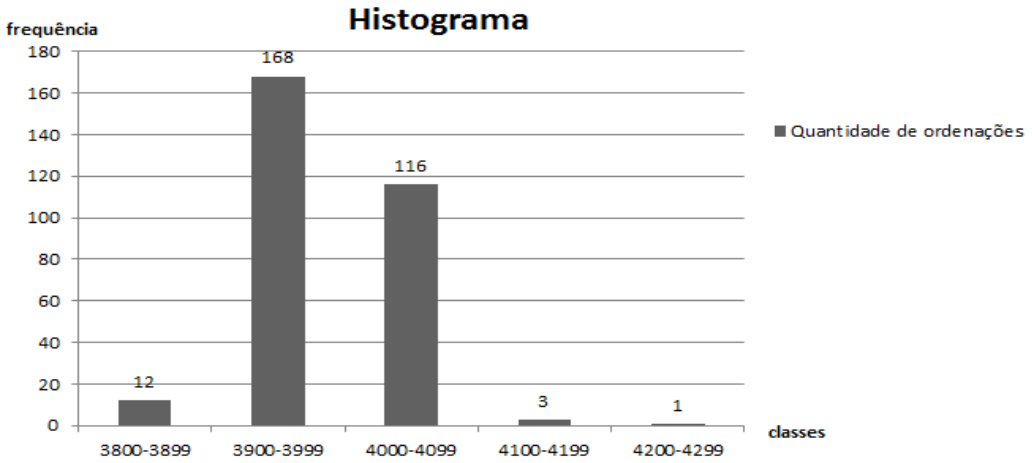

Figura 1. Histograma dos dados

Tabela 1. Valores de indicadores estatísticos para os tempos de ordenação

Indicadores Estatísticos	Valor
Média	3986,08 ms
Mediana	3990,50 ms
Moda	4013,00 ms
Variância	2238,96 ms²
Desvio Padrão	47,32 ms
Coeficiente de Variação	0,01
Mínimo	3873,00 ms
Máximo	4242,00 ms
Amplitude	369,00 ms
1º quartil	3959,00 ms
3º quartil	4014,75 ms
2º decil	3948,00 ms
4º decil	3978,00 ms
6º decil	3999,60 ms
30º centil	3965,00 ms
60º centil	4000,00 ms
90º centil	4033,00 ms

Referências

Donald Knuth. The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0. Section 5.2.1: Sorting by Insertion, pp. 80–105.

http://pt.wikipedia.org/wiki/Insertion_sort

Anexo*:

Fizemos o cálculo do intervalo de confiança da média de um conjunto de dados que foi gerado a partir da diferença do conjunto de dados usado no artigo e um gerado pelo algoritmo quicksort para os mesmos 300 vetores do experimento do artigo. Usaremos os resultados para fazer uma breve comparação do desempenho dos 2.

Dados	Valor
Média	3911,75 ms
N	300
S^2	4800,62 ms^2
S	69,29 ms
t student	1,64
Raiz de N	17,32
Erro ((t*S)/sqrt(N))	6,58 ms
Limite inferior	3905,17 ms
Limite superior	3918,33 ms

Visto que o intervalo foi (3905.17 , 3918.33), a ausência do 0 no intervalo indica que o desempenho do quicksort é superior.

*não faz parte do artigo.