

# COMPARAÇÃO DO TEMPO DE EXECUÇÃO DE ALGORITMOS DE ORDENAÇÃO DE UM VETOR

PRADELA, Izaura Pereira<sup>1</sup>, OLIVEIRA, Lucineida Nara de Andrade<sup>1</sup>, QUEIROZ, Marcela Dantas<sup>1</sup>,  
PARREIRA JÚNIOR, Walteno Martins<sup>2</sup>

<sup>1</sup>Graduandas do Curso de Engenharia de Computação da UEMG - campus de Ituiutaba (UEMG-FEIT-ISEPI)  
email: izaupradela@bol.com.br, lucineida\_nara@hotmail.com, cellysindy@hotmail.com

<sup>2</sup>Professor dos cursos de Engenharia da Computação, Engenharia Elétrica e Sistema de Informação da UEMG -  
campus de Ituiutaba (UEMG-FEIT-ISEPI) – email: walteno@ituiutaba.uemg.br.

**Área FAPEMIG:** Ciência da Computação – 1.03.03.00-6

**Resumo:** *No desenvolvimento de softwares, é importante avaliar sua correção temporal. Este artigo apresenta análises de dois algoritmos diferentes que retornam uma lista com seus valores ordenados, compara o tempo de execução de cada um e faz uma análise da diferença de tempo entre eles.*

**Palavras-chave:** algoritmos de ordenação por inserção, shellsort e tempo de execução.

## 1. Introdução

Este artigo relata uma das atividades desenvolvidas no projeto de estudo e implementação de algoritmos de ordenação que está sendo desenvolvido e tem por objetivo observar o funcionamento destes algoritmos em várias situações.

Segundo Ziviani (2002, p.69), “*Os métodos de ordenação constituem um bom exemplo de como resolver problemas utilizando computadores. As técnicas de ordenação permitem apresentar um conjunto amplo de algoritmos distintos para resolver uma mesma tarefa. Dependendo da aplicação, cada algoritmo considerado possui uma vantagem particular sobre os outros algoritmos.*”.

Para entender o funcionamento dos algoritmos, foi pesquisado um conjunto de métodos de ordenação descritos na literatura. Nas palavras de Ziviani (2002, p.69) “*Ordenar consiste no processo de rearranjar um conjunto de objetos em uma ordem ascendente ou descendente. O objetivo principal da ordenação é facilitar a recuperação posterior de itens do conjunto ordenado.*”.

Foram feitas comparações de tempos de execução em dois algoritmos de ordenação de vetores, cuja implementação foi realizada em Linguagem C. Foram utilizados algoritmos de ordenação interna em um conjunto pequeno de dados porque o processo de ordenação pode ser realizado inteiramente na memória principal.

Um dos critérios para a escolha de um algoritmo de ordenação em relação a outro é o tempo gasto para ordenar um arquivo, outro critério é o consumo de memória que necessita para executar o serviço. Neste caso, o interesse está no tempo utilizado.

## 2. Desenvolvimento

Os algoritmos implementados na linguagem C para comparação foram: Ordenação por inserção e o Shellsort. Segundo Celso Moraes (2001 p.226), O algoritmo shellsort é considerado uma extensão do

algoritmo de inserção, pois a diferença reside apenas no número de seguimentos do vetor usado para o processo de classificação. Deste modo estão sendo comparados dois métodos com o mesmo princípio.

### 2.1- Método de ordenação por Inserção

O método de ordenação por inserção troca itens adjacentes quando está procurando o ponto de inserção na seqüência destino. Se o menor item estiver na posição mais à direita no vetor então o número de comparações e movimentações é igual a  $(n-1)$  para encontrar o seu ponto de inserção. Assumindo que todas as permutações de  $n$  são igualmente prováveis para o caso médio, onde os itens estão distribuídos randomicamente, logo, o número de comparações é igual a  $M(n) = n^2/4 + 11n/4 - 3$ . Assim, a complexidade do algoritmo é  $C(n^2)$ .

### 2.2- Método de ordenação Shellsort

O método de ordenação shellsort contorna o problema método de inserção de encontrar seu ponto de inserção, permitindo trocas de registros que estão distantes um do outro. Os itens que estão separados  $h$  posições são re-arranjados de tal forma que todo  $h$ -ésimo item leva a uma seqüência ordenada. O valor de  $h$  pode ser estipulada para um determinada implementação, somente não podem ser utilizados valores múltiplos, sob pena de perda de performance. Tal seqüência é dita estar ordenada. Para a seqüência de incrementos utilizada no algoritmo existem duas conjecturas para o número de comparações:  $C(n) = O(n^{1,25})$  e  $C(n) = O(n \ln n^2)$ , e isto ainda não foi determinada, segundo Ziviani (2002, p.77) por ter alguns problemas matemáticos complexos.

A tabela 1 mostra a seqüência de ordenação usando os incrementos: 4, 2 e 1 como valores de  $h$ .

Posição:	1	2	3	4	5	6
Chave inicial	O	R	D	E	N	A
$h = 4$	N	A	D	E	O	R
$h = 2$	D	A	N	E	O	R
$h = 1$	A	D	E	N	O	R

Tabela 1 – Seqüência de ordenação usando os incrementos (fonte: ZIVIANI, 2002)

Segundo Parreira Junior (2006, p.11), A seqüência de execução pode ser descrita nos seguintes passos. Na primeira passada ( $h=4$ ), a letra O é comparada com a letra N (posições 1 e 5) e trocados, a seguir o R é comparado com o A (posições 2 e 6) e trocados. Na segunda passada ( $h=2$ ), as letras N, D e O nas posições 1, 3 e 5 são re-arranjadas para resultar em D, N e O nestas mesmas posições; da mesma forma as letras A, E e R nas posições 2, 4 e 6 são comparados e mantidos nos seus lugares. A última passada ( $h=1$ ) corresponde ao algoritmo de inserção, entretanto nenhum item tem que se mover para posições muito distantes.

Ao executarmos os algoritmos, entramos com o vetor de números inteiros ordenados de forma decrescente, aumentando assim a complexidade do algoritmo exigindo mais comparações em seu processamento de ordenação.

## 3. Resultados obtidos

Para desenvolver a experiência, os dois métodos foram implementados e testados para várias quantidades de itens e seus tempos foram anotados para a posterior análise dos dados.

### 3.1-Apresentação dos resultados da ordenação por inserção:

Tamanho do vetor	Média (segundos)	Tempo 1	Tempo 2	Tempo 3
10	43,33s	43s	44s	43s
20	183,67s	187s	187s	177s
30	238,3s	312s	220s	183s
40	767s	761s	770s	770s

Tabela 2 - tempo de execução do método por inserção

### 3.2- Apresentação dos resultados da ordenação shellsort:

Tamanho do vetor	Média (segundos)	Tempo 1	Tempo 2	Tempo 3
10	30s	31s	30s	29s
20	74s	74s	74s	74s
30	133,33	137s	136s	136s
40	236,33s	235s	236s	238s

Tabela 3 - tempo de execução do método shellsort

Pode-se observar que os tempos encontrados na execução do método por inserção (tabela 2) são maiores que os correspondentes tempos do método shellsort (tabela 3) para qualquer tamanho do vetor. Os valores serem maiores no método por inserção já era esperado, conforme apresentado anteriormente, pois a complexidade do método por inserção é quadrática e do shellsort, mesmo não sendo determinada é menor que isto, ficando entre  $O(n^{1,25})$  e  $O(n \ln n)^2$ .

## 4. Conclusão

Observa-se que o tempo de execução do algoritmo shellsort é menor que o tempo de execução do algoritmo de ordenação por inserção para os vetores de todos os tamanhos, como era esperado, pois a sua complexidade é menor, mesmo não sendo demonstrada. Não temos como calcular o número certo de comparações em cada operação por ter não ter uma regra geral para qualquer quantidade de elementos (n). Os cálculos são de difícil entendimento, mas serão objetos de novos estudos.

Segundo Ziviani (2002, p. 77), o método de ordenação shellsort é uma ótima opção para ordenação de arquivos com uma quantidade moderada de registros, na casa de 5000, até porque sua implementação simples e requer pouca programação.

## 5. Bibliografia

MORAES, Celso Roberto. **Estruturas de dados e algoritmos:** uma abordagem didática. São Paulo: Berkeley Brasil, 2001.

PARREIRA JÚNIOR, Walteno Martins. **Métodos de ordenação (Apostila).** Ituiutaba: FEIT-UEMG, 2006.

ZIVIANI, Nívio. **Projeto de algoritmos:** com implementação em Pascal e C. São Paulo: Pioneira – Thonson Learning, 2002.

### **Para Referencia do Artigo:**

PRADELA, Izaura Pereira, OLIVEIRA, Lucineida Nara de Andrade, QUEIROZ, Marcela Dantas & PARREIRA JÚNIOR, Walteno M. Comparação do tempo de execução de algoritmos de ordenação de um vetor. IN: Seminário de Iniciação Científica e extensão da UEMG, 10, 2008, Divinópolis (MG). **Anais do 10 Seminário da UEMG**. Divinópolis: UEMG e FUNED. 2008. CD-ROM. ISSN: 1983-9693. Disponível em <[www.waltenomartins.com.br/artigos](http://www.waltenomartins.com.br/artigos)>