

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

Izaura Pereira Pradela¹, Lucineida Nara de Andrade Oliveira¹, Marcela Dantas Queiroz¹,
Walteno Martins Parreira Júnior²

¹Graduandas do Curso de Engenharia de Computação da UEMG - campus de Ituiutaba (UEMG-FEIT-ISEPI)
email: izaurapradela@bol.com.br, lucineida_nara@hotmail.com, cellysindy@hotmail.com

²Professor 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.

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

1. Introdução

Comparar o tempo de execução de dois algoritmos diferentes de ordenação e analisar a diferença de tempo entre eles

2. Fundamentações Teóricas

2.1. Técnicas de Ordenação

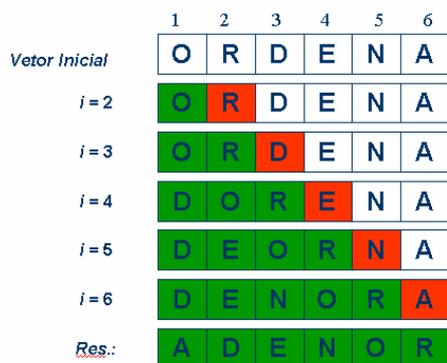
As técnicas de ordenação permitem apresentar um conjunto amplo de algoritmos distintos para resolver uma mesma tarefa. Dependendo da aplicação, cada algoritmo considerado possui uma vantagem particular sobre os outros algoritmos.

2.2. Métodos de Ordenação

Os métodos de ordenação constituem um bom exemplo de como resolver problemas utilizando computadores.

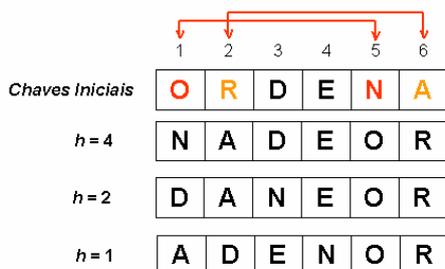
Os algoritmos de ordenação implementados na linguagem C para comparação foram: a) por inserção e b) Shellsort

Ordenação por inserção:



A complexidade deste método é $O(n^2)$ para os casos médio e pior e o número de comparações para o caso médio é $C(n) = \frac{1}{4}(n^2 + 11n) - 3$.

Shellsort:



Existem duas conjecturas para o número de comparações desse algoritmo: $C(n) = O(n^{1,25})$ e $C(n) = O(n \ln n)$, e isto ainda não foi determinada, segundo Ziviani (2002, p.77) por ter alguns problemas matemáticos complexos.

3. Desenvolvimento

Os algoritmos foram implementados na Linguagem C, compilados no TurboC e executados várias vezes no mesmo computador.

4. 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.

4.1 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

4.2 Análise do algoritmo de ordenação por inserção:

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

5. 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.

6. 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 – Thomson Learning, 2002.