Elias Flávio de Paiva

Algoritmos e Estrutura de Dados: Algoritmos de ordenação

* Implementação

Nesta parte foram utilizados vetores pequenos gerados aleatoriamente com tamanhos definidos para validar os algoritmos.

A implementação de cada algoritmo está disponível no GitHub pelo link: https://github.com/eliasfpaiva/Trabalhos\_PUC/blob/master/AED/Algoritmos%20de%20ordenação/Algoritmos-Ordenacao/Algoritmos-Ordenacao/Classes/Algoritmos.cs

* Análise de Desempenho Experimental

Códigos utilizados para realizar as medições disponíveis no GitHub no link: https://github.com/eliasfpaiva/Trabalhos\_PUC/blob/master/AED/Algoritmos%20de%20ordenação/Algoritmos-Ordenacao/Algoritmos-Ordenacao/Program.cs

Seguem os gráficos obtidos:

* + Ordenação com vetor desordenado
  + Ordenação com vetor em ordem crescente

* + Ordenação com vetor em ordem decrescente
* Análise de os Resultados

Em todos os testes os algoritmos MergeSort e QuickSort apresentaram os resultados mais satisfatórios independentemente da ordem em que se encontravam as informações de entrada, o que não é verdade para o outros algoritmos testados onde é visível a variação de eficiência.

Todos os algoritmos são capazes realizar a ordenação, mas, existe uma considerável diferença de desempenho entre eles.

* Conclusão

É interessante atestar o funcionamento de ferramentas tão importantes como algoritmos de ordenação. No dia-a-dia, e principalmente em testes em sala de aula não faz muita diferença qual algoritmo usamos, pois, geralmente trabalhamos com quantidades mínimas de dados, mas, com uma quantidade mais considerável, como foi o caso diferenças ficam evidentes e a escolha de um ou outro se mostra importante.

* Bibliografia

LEAL, Gislaine Camila Lapasini, Algoritmos e Lógica de Programação II, Núcleo de Educação a Distância Unicesumar – Maringá-Pr, 2016.

SOUZA, Kleber Jacques F. de, Aula 06 – Ordenação Interna – Bolha, Seleção e Inserção.

SOUZA, Kleber Jacques F. de, Aula 07 – Ordenação Interna – MergeSort e QuickSort.