

UMA FERRAMENTA GRÁFICA PARA O ESTUDO E ANÁLISE COMPORTAMENTAL DE ALGORITMOS DE ORDENAÇÃO

COLARES, V. M. M.; OLIVEIRA, A. S.; GONÇALVES, J. J.; CORRÊA, M. M.

Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSul) – Bagé – RS – Brasil
v.colares@outlook.com; arthursdo@outlook.com; juliajardimg@yahoo.com.br;
marcelcorrea@ifsul.edu.br

RESUMO

Este trabalho propõe uma ferramenta para o estudo de algoritmos de ordenação através da apresentação gráfica do processo de ordenação. A ferramenta permite a visualização das características comportamentais de cada algoritmo, além de fornecer dados de desempenho. Palavras-chave: Ordenação; *Bubble Sort*; *Selection Sort*; *Insertion Sort*; *Cocktail Sort*.

1 INTRODUÇÃO

A ordenação é o problema computacional de se colocar um conjunto de elementos em uma determinada ordem. Os algoritmos de ordenação são aqueles que exploram características destes conjuntos de dados para que se chegue em uma ordem desejada – como por exemplo, ordem crescente, decrescente e lexicográfica.

Os diversos algoritmos de ordenação se diferem principalmente pela eficiência ou complexidade, sendo os algoritmos menos eficientes os de mais fácil compreensão. Em termos de complexidade, os algoritmos de ordenação são classificados pela notação assintótica (CORMEN, 2012), e nesta notação os algoritmos mais simples são os de ordem quadrática, ou $O(n^2)$ em notação *Big-O*.

A ferramenta discutida neste trabalho tem como objetivo auxiliar professores e alunos no estudo de algoritmos de ordenação através da apresentação gráfica do processo de ordenação passo a passo – ou seja, destacando cada operação de comparação e de troca entre os elementos que estão sendo ordenados. Além disso, a ferramenta fornece ao usuário dados do custo da ordenação em termos de número de operações realizadas, permitindo assim a comparação de desempenho entre os diferentes algoritmos para um mesmo conjunto de dados.

Tabela 1 – Algoritmos suportados pela ferramenta.

<i>Bubble Sort</i>	<i>Selection Sort</i>
<pre>para i=0:n-1 passo 1 para j=n-1:i+1 passo -1 se V[j-1]>V[j] troca V[j-1],V[j]</pre>	<pre>para i=0:n-1 passo 1 k = i para j=i+1:n-1 passo 1 se V[j]<V[k] k = j troca V[i],V[k]</pre>
<i>Insertion Sort</i>	<i>Cocktail Sort</i>
<pre>para i=1:n-1 passo 1 para j=i:1 passo -1 se V[j-1]>V[j] troca V[j-1],V[j] senao pare</pre>	<pre>ini = n-1, fim = 0 enquanto fim>ini para i:fim passo 1 se V[i],V[i+1] troca V[i],V[i+1] fim = fim-1 para i:ini passo -1 se V[i-1],V[i] troca V[i-1],V[i] ini = ini+1</pre>

Nesta versão da ferramenta (1.4a de 22/09/2015) os algoritmos implementados foram quatro conhecidos algoritmos de ordem quadrática: *Bubble Sort*, *Selection Sort*, *Insertion Sort* e *Cocktail Sort*. Estes algoritmos estão descritos em pseudocódigo na Tabela 1, sendo n a quantidade de elementos e V o *array* de elementos com posições de 0 até $n-1$.

2 METODOLOGIA

A tecnologia escolhida para implementação da ferramenta foi o Portugol Studio (UNIVALI, 2015). Esta escolha foi motivada pelo fato de a linguagem de programação, ambiente de desenvolvimento e documentação serem todos em português, facilitando o rápido aprendizado por parte dos alunos. Outro grande facilitador foi o fato de a linguagem fornecer uma biblioteca nativa para a manipulação de gráficos 2D.

A ferramenta foi construída para funcionar em três etapas bem definidas, sendo elas: (1) a entrada dos parâmetros por parte do usuário – escolha do algoritmo, tamanho do conjunto de dados e velocidade da animação –, (2) execução da ordenação com apresentação gráfica do processo passo a passo, e (3) apresentação dos dados numéricos de desempenho do algoritmo.

A etapa de ordenação consiste primeiramente na criação do conjunto de dados seguida do embaralhamento para que esta coleção assuma uma ordem inicial aleatória. Para este embaralhamento foi implementado o algoritmo *Fisher-Yates*, com o qual todas as possíveis ordens que o conjunto pode assumir são igualmente prováveis. Após a criação do conjunto a ferramenta inicia o modo gráfico, no qual o conjunto de dados é representado por uma sequência de barras verticais brancas de altura proporcional ao valor do dado. A animação gera um novo quadro para cada operação de comparação entre um par de elementos, destacando este par de barras em vermelho quando o algoritmo efetua a troca de posição destas barras, ou em verde quando a posição das barras é mantida.

Um exemplo do funcionamento do modo gráfico para um conjunto de 75 elementos e para o algoritmo *Insertion Sort* está ilustrado na Figura 1, onde pode ser visto que no quadro capturado houve uma troca de posição entre as barras vermelhas, e pode ser visto também que o conjunto se encontra parcialmente ordenado.



Figura 1 – Captura de um quadro durante a execução do algoritmo *Insertion Sort* sobre um conjunto de 75 elementos.

3 RESULTADOS E DISCUSSÃO

Na prática, o uso em testes da ferramenta atendeu todos os objetivos esperados. Através dela é possível o entendimento de como cada algoritmo de ordenação opera e também de que não existe um algoritmo ideal para qualquer situação. As vantagens e desvantagens de cada algoritmo ficam perceptíveis, especialmente quando ocorrem alterações no tamanho do conjunto e na distribuição inicial dos elementos.

As mais importantes características de um algoritmo de ordenação são: Estabilidade, adaptabilidade, complexidade de espaço e complexidade de tempo. Um algoritmo estável é aquele que preserva a ordem relativa de elementos de mesmo valor, já um algoritmo adaptativo é aquele que fica mais eficiente quando o conjunto já está parcialmente ordenado, enquanto as complexidades de espaço e de tempo se referem respectivamente ao espaço de memória necessário e ao tempo de execução em relação ao tamanho do conjunto. A Tabela 2 descreve as características de cada um dos algoritmos suportados pela ferramenta.

Tabela 2 – Algoritmos de ordenação e suas principais características.

	Bubble Sort	Selection Sort	Insertion Sort	Cocktail Sort
Estável	Sim	Não	Sim	Sim
Adaptativo	Não	Não	Sim	Não
Complexidade (pior caso)	$O(n^2)$ comp., $O(n^2)$ trocas	$O(n^2)$ comp., $O(n)$ trocas	$O(n^2)$ comp., $O(n^2)$ trocas	$O(n^2)$ comp., $O(n^2)$ trocas
Complexidade espacial (pior caso)	$O(n)$ conjunto, $O(1)$ auxiliar	$O(n)$ conjunto, $O(1)$ auxiliar	$O(n)$ conjunto, $O(1)$ auxiliar	$O(n)$ conjunto, $O(1)$ auxiliar

Todas estas características podem ser facilmente vistas através da ferramenta proposta. Por exemplo, o *Selection Sort* não é estável por comparar e realizar trocas entre pares de elementos que não são adjacentes. No entanto, por conta disso este mesmo algoritmo é capaz de realizar um número de trocas linear em relação ao tamanho do conjunto, enquanto os outros realizam um número de trocas que tende ao quadrado do tamanho do conjunto.

4 CONCLUSÕES

Este trabalho propôs uma ferramenta para o estudo de algoritmos de ordenação através da apresentação gráfica do processo de ordenação etapa por etapa. A ferramenta atendeu todos objetivos iniciais dos autores em sua fase de testes.

Como trabalhos futuros se planeja a inserção de algoritmos de ordenação recursivos de ordem linear-logarítmica, ou $O(n \cdot \log n)$ em notação *Big-O*. Se planeja também a inserção de recursos mais robustos, como a possibilidade de execução de mais de um algoritmo de ordenação lado a lado em paralelo, facilitando assim a comparação direta entre algoritmos através do modo gráfico.

REFERÊNCIAS

CORMEN, T.; LEISERSON, C.; RIVEST, R.; STEIN C. **Algoritmos – Teoria e Prática**, 3.ed. Elsevier, 2012.

Portugol Studio. Disponível em: <<http://lite.acad.univali.br/portugol/>>. Acesso em 30 set. 2015.