



# MÉTODOS DE ORDENAÇÃO DE DADOS: UMA ANÁLISE PRÁTICA I

DA ROSA, Luciano<sup>1</sup>; ANTONIAZZI, Rodrigo Luiz<sup>2</sup>

**Palavras-Chave:** Ordenação. Métodos. Algoritmos.

## Introdução

Ordenação de dados são operações computacionais básicas indispensáveis e comumente utilizadas com o intuito de facilitar a compreensão e o rápido acesso á variados tipos de dados manipulados por aplicações. Existem diversos métodos dedicados ao processo de ordenação de dados computacionais, alguns dos quais abordam uma grande variedade de importantes técnicas existentes em algoritmos com fins diversos, sendo que o estudo e a análise de seu comportamento na teoria ou prática possuem relevância histórica (COLLINGS, 1988), (CORMEN, 2002).

O foco deste artigo é voltado para uma análise prática e comparativa entre os métodos de ordenação de dados: Bobble Sort, Insertion Sort, Selection Sort, Shell Sort e Quick Sort, submetendo-os a entradas de dados decrescentes (Invertidas), embaralhadas com repetições, embaralhadas sem repetições e levando em consideração fatores como a medida do tempo de execução do método e o numero de comparações e trocas necessárias para completar a ordenação.

## Metodologia

Na ordenação Booble Sort o elemento na posição  $i$  é comparado com o elemento na posição  $i+1$ , de forma crescente ou decrescente caso a condição da comparação for aceita então uma variável auxiliar irá guardar o valor de  $i+1$  que então recebe o valor de  $i$ , então  $i$  recebe o valor contido na variável auxiliar, ocorrendo então a troca de posições entre os elementos (ASCENCIO, 2010), (COLLINGS, 1988).

O método de ordenação Insertion Sort trabalha subdividindo a entrada de dados em duas listas, uma delas receberá as permutações de chaves já ordenadas no processo de ordenação, e a outra lista será decrementada ao longo do processo de ordenação. Será utilizado um elemento  $i$  chamado eleito, que permeará entremeio as duas listas, inicialmente o primeiro elemento da

---

<sup>1</sup> Acadêmico do Curso de Ciência da Computação UNICRUZ

<sup>2</sup> Docente do Curso de Ciência da Computação UNICRUZ



entrada de dados é eleito e em seguida o conteúdo do elemento é então comparado com os demais elementos do vetor  $i+1$  até  $i+n$  (ASCENCIO, 2010), (COLLINGS, 1988).

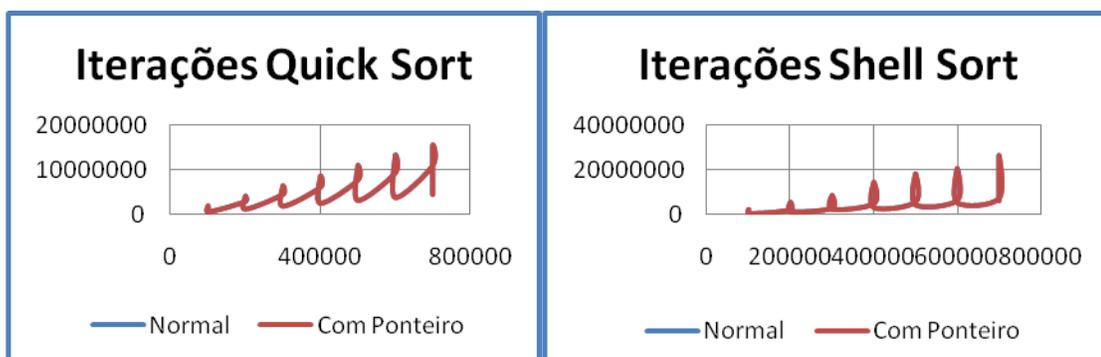
No método Selection Sort o elemento de menor valor entre os elementos da lista é selecionado e em seguida é feita a troca deste elemento com o primeiro elemento  $i$ , em seguida são selecionados e permutados os elementos de menor valor para as posições  $i+1$ ,  $i+2$ ... $i+n$  (ASCENCIO, 2010), (COLLINGS, 1988).

O método Quick Sort consiste basicamente em dividir o arranjo inicial de elementos em sucessivos sub-arranjos, sendo que a ordenação dos elementos se dará à medida que ocorram as subdivisões. Inicialmente é escolhido um elemento que se encontra na posição central do arranjo inicial, subdividindo o arranjo em dois sub-arranjos, de modo que por fim todos os elementos do sub-arranjo da esquerda contêm números menores ou iguais ao sub-arranjo da direita (ASCENCIO, 2010), (COLLINGS, 1988), (CORMEN, 2002).

No método Shell Sort o valor de  $h$  é quebrado em sub-partições ou valores menores e seu valor é atribuído a um auxiliar que então determina a distância  $h$  que deverá ser comparada entre os elementos do arranjo, após cada laço de repetição o valor do auxiliar é decrementado e a distância de  $h$  é recuada determinando comparações de elementos sucessivamente mais próximos (SEEDGEWICK, 1992).

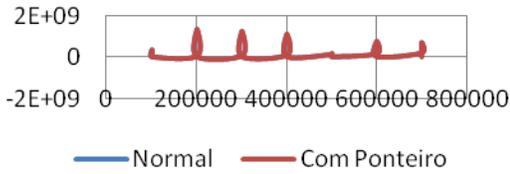
## Resultados e Discussões

Para um melhor entendimento sobre o funcionamento dos métodos e os fatores influentes nas questões de desempenho, abaixo se tem gráficos representativos exibindo a progressividade de cada método analisado.

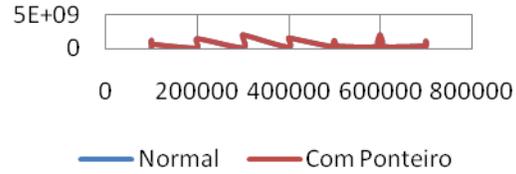




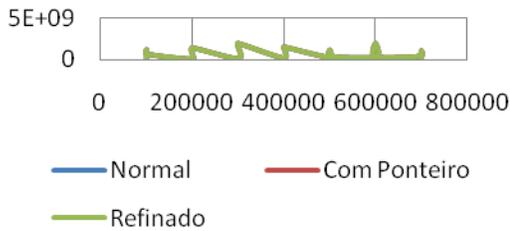
### Iterações Selection Sort



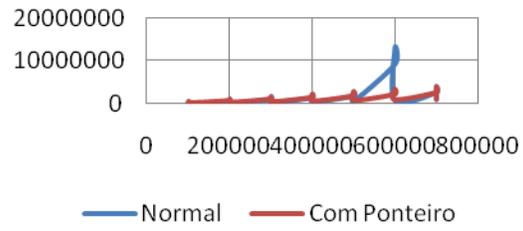
### Iterações Insertion Sort



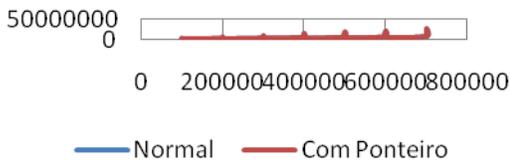
### Iterações Booble Sort



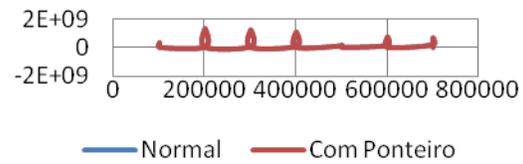
### Trocas Quick Sort



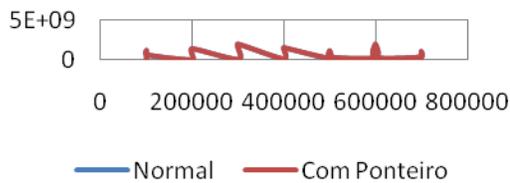
### Trocas Shell Sort



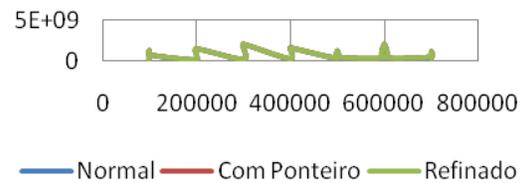
### Trocas Selection Sort



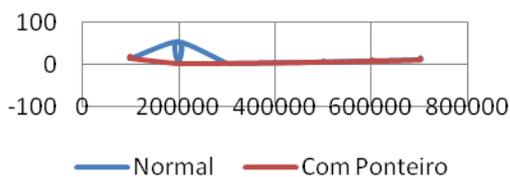
### Trocas Insertion Sort



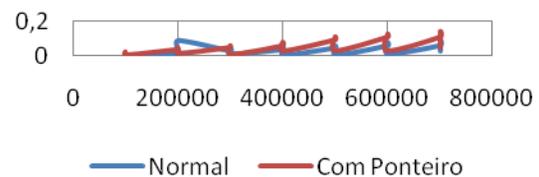
### Trocas Booble Sort



### Tempo Selection Sort



### Tempo Quick Sort



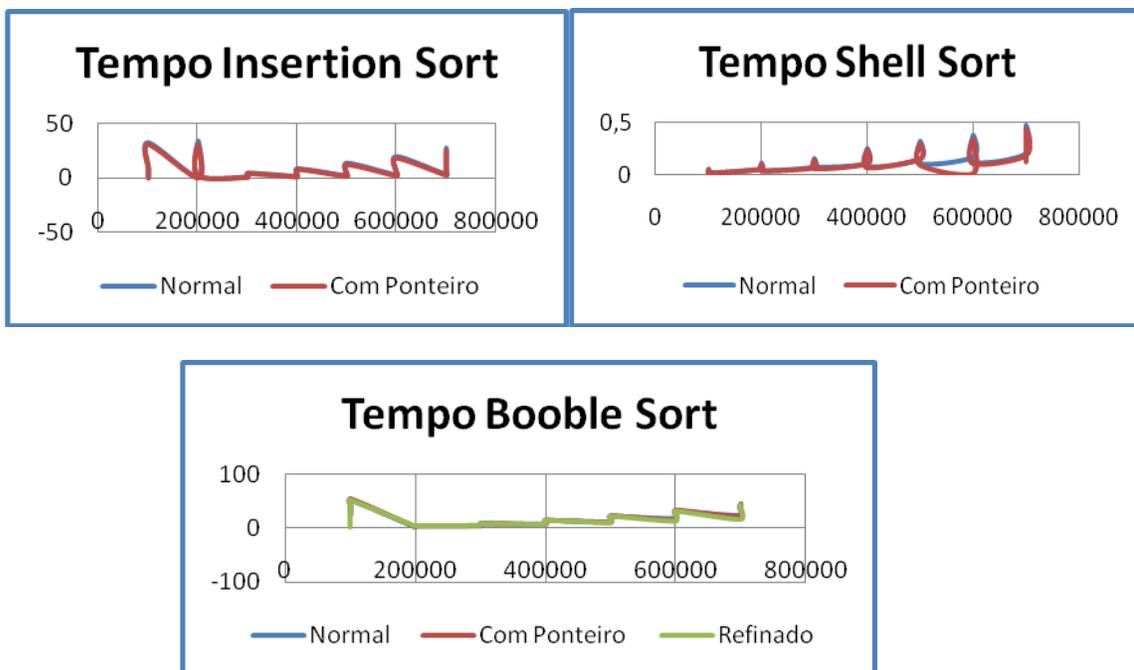


Figura 1: Representação Gráfica da análise da execução dos métodos de ordenação;

## Conclusão

Em relação às iterações o método Quick Sort mostrou um crescimento mais estabilizado e menor ao longo das escalas de entrada, o método com mais índices de trocas foi o método bolha, convergindo e tendo um ganho rápido. Nas trocas de posições o método Quick Sort e o Shell obtiveram um resultado plausível com poucas trocas em relação á escala de entrada, já o Booble o Insertion cresceram demasiadamente resultando em um tempo de execução mais demorado. Na análise do tempo de execução observa-se que o método Quick Sort e o Shell Sort obtiveram tempos de execução espetaculares em relação aos tempos de execução desenvolvidos pelos demais métodos e em relação ás escalas de entrada, chegando a segundos e a milésimos de segundo.

## Referências

- ASCENCIO, Ana Fernanda Gomes. **Estruturas de Dados: Algoritmos, análise da complexidade em JAVA e C/C++**. São Paulo: Pearson Prentice, 2012.
- SEDGEWICK, Robert. **Algorithms in C++**. Princeton University: Addison-Wesley Publishing Company, 1992.
- CORMEN, Tomas H. **Algoritmos: Teoria e Prática**. Rio de Janeiro: Campus, 2002.
- COLLINS, William J. **Programação Estruturada com estudos de caso em PASCAL**. São Paulo: McGraw-Hill, 1998.