

# **Pesquisa e Ordenação de Dados “A”: Trabalho 1 – Algoritmo de Ordenação Externa**

**Gustavo Arrua Fantinel**

`gafantinel@inf.ufsm.br`

## **1. Introdução**

O presente trabalho tinha como proposta o desenvolvimento de um software capaz de receber um arquivo .txt contendo palavras e utilizar uma implementação de Mergesort ou Quicksort de maneira externa, afim de que se solucionasse a limitação imposta de memória interna reduzida. O algoritmo escolhido foi o Mergesort e a linguagem de programação utilizada foi Java, dado que estou buscando me aprofundar nessa linguagem.

## **2. Algoritmo**

Antes da elaboração propriamente dita do programa, estudei por diversas fontes sobre o algoritmo de Mergesort e sobre os algoritmos de ordenação externa. As notas de aula cedidas pelo professor foram importantes nesse processo. Estudei, também a respeito das diversas bibliotecas do Java, escolhendo aquelas que seriam mais úteis na implementação do código.

O algoritmo assim se comporta:

O arquivo de texto é carregado dentro do método status e nele é “quebrado” em tokens.

Os tokens, então, alimentam um vetor de strings enquanto a memória disponível for suficiente.

Quando a memória é excedida, o vetor é ordenado alfabeticamente e gravado em um arquivo temporário.

Esse ciclo se repete até que todos os tokens do arquivo original estejam distribuídos ordenadamente em arquivos temporários.

Logo, o método status retornará um valor false, fazendo com que o loop da main comece a iterar.

As iterações chamam o método merge que lê cada arquivo temporário através de buffers, ordenando-os e gravando-os em novos arquivos temporários.

Ao final dessas iterações, temos um arquivo ordenado alfabeticamente.

### **3. Aplicações**

O algoritmo de Mergesort, quando implementado utilizando-se de ordenação externa à memória primária tem inúmeras aplicações no “mundo real”. Todos os dias, diversas empresas, desde pequenas até multinacionais, precisam ordenar trilhões de informações e é nesse cenário que algoritmos como o Mergesort, dentre outros, se destacam. São soluções econômicas no quesito de custo computacional e ainda entregam resultados relativamente rápidos quando comparados a outros métodos de ordenação de dados.

### **Referências**

LIMA, João Vicente Ferreira. **Notas de Aula**. 2018. Disponível em: <<https://github.com/joao-lima/notas-de-aula-POD-UFSM>>. Acesso em 14 de maio de 2018.