

	<b>DATA DE ENTREGA:</b>
<b>CURSO:</b> Análise e Desenvolvimento de Sistemas	<b>NOTA:</b>
<b>DISCIPLINA:</b> Estrutura de Dados	
<b>PROFESSOR:</b> Luiz Mário Lustosa Pascoal	
<b>ALUNO:</b>	

## ESPECIFICAÇÃO DO TRABALHO SOBRE BUSCA E ORDENAÇÃO

### Grupos de no máximo 3 pessoas

Com base nos algoritmos de busca e ordenação trabalhados em sala de aula, desenvolva um programa em Java que seja capaz de ler um arquivo .txt e realize a ordenação das palavras presentes neste arquivo em ordem crescente de acordo com o tamanho delas, salvando o resultado no arquivo saída.txt, conforme o exemplo abaixo.

Entrada.txt	Saída.txt / tamanho da palavra	
Uva	Uva	3
Banana	Pêra	4
Maçã	Maçã	4
Pêra	Banana	6
Damasco	Damasco	7

Para testar o algoritmo, vocês devem entrar no site: <https://github.com/titoBouzout/Dictionaries> que apresenta uma coletânea com diversos dicionários de idiomas ao redor do mundo. Os dicionários são armazenados nos arquivos .dic, cuja a primeira linha indica a quantidade de palavras presentes no dicionário e as demais contém as palavras em ordem alfabética. Use a informação da quantidade de palavras para criar um **vetor** do tipo **String** com o tamanho informado e armazene cada palavra do arquivo neste vetor de forma correta.

Em seguida, ordene o vetor de acordo com o tamanho das palavras presentes nele utilizando os algoritmos **SelectionSort** ou **BubbleSort**, **InsertionSort**, **MergeSort** e **QuickSort** (totalizando 4 algoritmos de ordenação). Ao final do processo de ordenação, o programa deverá computar o tempo em **milissegundos** ou **nanossegundos** (dependendo do tamanho do vetor) levado pelo método para realizar a ordenação.

- Obs. Veja:

- o `long tempo = System.nanoTime();` // recebe o tempo de máquina em nanossegundos.
- o `long tempo = System.currentTimeMillis();` // recebe o tempo de máquina em milissegundos

Por fim, o programa também deve realizar a busca de uma palavra qualquer no dicionário utilizando os métodos de busca Sequencial e Binária, computando também o tempo levado pelos algoritmos para realização desta tarefa. Teste o programa utilizando pelo menos 3 dicionários de tamanhos diferentes, onde obrigatoriamente a língua **Portuguesa (Portuguese (Brazilian).txt)** deverá ser testada.

**Ao final gere também um relatório descritivo em PDF sobre a execução e observações realizadas durante o desenvolvimento do trabalho acerca de cada algoritmo desenvolvido, explicitando o funcionamento de cada algoritmo bem como a realização de uma análise sobre quais algoritmos se comportaram melhor na resolução do problema. DESAFIO: Encontrar versões que executem paralelamente (com threads) cada um dos algoritmos de ordenação, testar se há melhoria no desempenho. Descreva seu funcionamento.**

**Bom trabalho!!**