Busca e Ordenação

Alunos:
Dhionatã Carlos Vieira
Murillo Maciel
Vitor de Oliveira Birindiba

Relatório

Classe principal :

Está classe possui o método “main”, o qual comporta e faz uso dos métodos das classes complementares, as quais são:

- “Ordenacao”;
- “Busca”;
- ”ArquivoHandler”;
- ”Saida”.

A classe também faz uso de “import java.util.Scanner”, para inserção de dados e/ou informações para o funcionamento da mesma.

Estrutura da classe:

Esta começa por iniciar dois (2) contadores, um em nano segundos e o outro em milissegundos, respectivamente.

São feitos quatro (4) vetores, sendo três (3) para a leitura dos arquivos .txt utilizando a classe “ArquivoHandler” e um (1) auxiliares.

A **1ª** etapa da execução da classe é feita pela pergunta de qual dicionário a ser usado, tendo as opções dos seguintes:

- English
- English (Canadian)
- English (South African)

A **2ª** etapa da execução da classe é feita com a escolha de um método de ordenação, executada pela classe “Ordenacao”, podendo ser feito das seguintes maneiras:

- QuickSort
- MergeSort
- BubbleSort
- InsertionSort
- SelectionSort

A **3ª** fase de execução é a busca por uma palavra, a qual faz o uso da classe “Busca”, o qual são dadas duas (2) opções de métodos de pesquisa dentro do vetor já ordenado anteriormente.

O **4º** estágio da execução é quando se é perguntado se o utilizador deseja salvar o vetor organizado por ordem de tamanho dentro do arquivo “saida.txt”, que faz uso da classe “Saida” para tal.

A **5ª** fase de execução é quando são exibidos o tempo gasto tanto em nanosegundos quanto em milissegundos.

O **6º** e último estágio é quando é dada a opção de se escolher outro dicionário (looping) no código, e reinício da classe.

Segurança:

Existem nas opções, verificações para garantir que o utilizador não faça o uso indevido das mesmas.

Classe de Busca:

A classe de busca funciona através de dois métodos, sequencial e binário. A busca sequencial procura o valor a partir de várias comparações sucessivas a partir do primeiro elemento ao último, até o momento que o valor desejado é encontrado, ou o vetor acabe.

A busca binária utiliza do método “dividir e conquistar”, onde ela vai dividindo o vetor várias vezes, e depois vai comparando seus valores com a função

“equalsIgnoreCase()”, que por meio de um “for” faz uso de uma busca sequencial apenas nas palavras contendo o mesmo tamanho da palavra que está sendo procurada.

Após os primeiros testes de execução, a busca binária demonstrou ser razoavelmente mais rápida que a busca sequencial.

Classe do ArquivoHandler:

A classe “ArquivoHandler” executa apenas uma função basicamente, ela lê uma String em formato .txt, que é recebida através do parâmetro de seu método, e o armazena em um vetor, para que este arquivo possa ser ordenado pela classe “Ordenacao”.

Classe de Saída:

A classe “saida” tem como função ler o arquivo.txt que está armazenado no vetor, e o reescrever em uma saída.txt, após ser ordenado pela classe “Ordenacao”, e enumerar o tamanho de cada String.

Classe Ordenação:

A classe “Ordenacao” utiliza de cinco métodos, são eles: QuickSort, MergeSort, BubbleSort, InsertionSort e SelectionSort. Em questão de velocidade QuickSort e MergeSort apresentaram melhor desempenho, InsertionSort e SelectionSort obtiveram desempenho mediano, abaixo de 5 segundos, já o BubbleSort provocou engasgos na máquina, demorando em média mais de 5 segundos para realizar a ordenação do vetor.