Universidade Federal do Maranhão
Departamento de Informática
Disciplina: ***Estrutura de Dados II 2022.1***
Prof.: ***João Dallyson Sousa de Almeida***

Atividade Prática 1 (Dupla)
Valor: 50% da 1ª Avaliação

1. DESCRIÇÃO

1) Implemente o algoritmo do **SelectSort, QuicSort, HeapSort e MergeSort** padrão apresentados em aula.
2) Implemente uma outra versão do algoritmo SelectSort. O passo de seleção do menor ou maior item deve ser modificado para selecionar o maior e menor na mesma iteração (verifique algoritmo minmax3 na nota de aula 2, apresentado em aula). Seguindo essa ideia, o algoritmo já organizará o menor e maior elemento a cada iteração.
3) **Usando o InsertSort para subarrays pequenos no MergeSort**. Podemos melhorar a maioria dos algoritmos recursivos tratando pequenos casos de maneira diferente, porque a recursão garante que o método será usado com frequência para casos pequenos, portanto, melhorias no tratamento deles levam a melhorias em todo o algoritmo. No caso da ordenação, sabemos que a ordenação por inserção (ou seleção) é simples e, portanto, provavelmente mais rápida do que a ordenação por mergesort para subvetores pequenos. Alternar para a ordenação por inserção para subvetores pequenos (tamanho 15 ou menor) melhorará o tempo de execução de uma implementação padrão do **MergeSort** em 10 a 15 por cento.
4) O usuário dever ter a opção de ordenar em ordem crescente ou decrescente em todos os algoritmos.
5) Compare a execução das 4 versões anteriores do MergeSort com os métodos de ordenação disponíveis no JDK do Java nas apis: java.util.Arrays e java.util.Collection

OBS:

1) Todos o métodos de ordenação deverão ordenar objetos utilizando *Generics* *(https://docs.oracle.com/javase/tutorial/java/generics/types.html).*
2) Você deverá utilizar o padrão de projeto *Strategy.* Utilize como referência a estrutura de código disponibilizada. Referência sobre o Strategy: *(http://www.ic.uff.br/~anselmo/cursos/ProjSoft/apresentacoes/PadraoStrategy.pdf).*
   O padrão Strategy pode ser implementado, de forma simplificada, utilizando Java Lambda Expressions (Java SE 8), exemplo: https://www.javacodegeeks.com/2016/01/java-8-lambda-expression-design-patterns-strategy-design-pattern.html
3) Você deverá enviar, também, um relatório mostrando e discutindo os resultados obtidos. Mostre prints da execução e dos resultados.

2. Descrição sobre os arquivos de dados

Execute experimentos com Vetores de tamanho pequeno (cerca de 1000), médio (>=100.000) e grande (>=1 milhão).

OBS: Você deverá gerar vetores de itens de vários tipos com valores de chaves aleatórias. Incluindo os seguintes:

a) Chave (tipo String com no mínimo 10 caracteres), valor (tipo Double)
b) Chave (tipo Double), valor (tipo String com no mínimo 10 caracteres)
c) Chave (tipo Double), valor (tipo Integer)

Analise o impacto dos diferentes tipos de entrada.

3. Análise dos resultados

A análise deve ser feita sobre o número de comparações, atribuições e tempo de execução dos algoritmos.

4. Entrega

- Código fonte do programa em JAVA (bem identado e comentado) utilizando os conceitos de Orientação a Objetos.
- Relatório com os resultados.
- Upload no SIGAA.

O Relatório deve apresentar:

1. Testes: apresentação dos testes realizados.
2. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
3. Referências: referências utilizadas no desenvolvimento do trabalho.