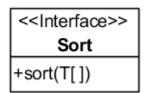
Trabalho 4 2022/1

Algoritmos de Ordenação (SORT)

Implemente um dos algoritmos de ordenação a seguir em uma classe cujo nome é o nome do algoritmo (Exemplo: Classe Insertion com método sort):

- 1. Block Sort
- 2. Bucket sort
- 3. Counting sort
- 4. Cubesort
- 5. Cycle sort
- 6. Gnome sort
- 7. Heapsort
- 8. Insertion sort
- 9. Introsort
- 10. Merge sort
- 11. Quick sort
- 12. Radix sort
- 13. Selection sort
- 14. Shell sort
- 15. Timsort
- 16. Tree Sort

Sua classe deve realizar a interface Sort, especificada abaixo:



Além da implementação, a equipe deve elaborar um vídeo explicativo sobre o algoritmo e sua implementação (em torno de 10 min), que deve ficar disponível no Youtube. Na aula de 27/06 assistiremos alguns vídeos (sorteados). Também faremos testes com todos os algoritmos para verificar sua eficiência.

Os algoritmos e códigos estão amplamente disponíveis na literatura e internet. Portanto, pesquise e desenvolva seu trabalho. Reforçando que não poderá ser utilizada nenhuma classe da JCF.

Coloque sua classe num pacote chamado 'ordenacao' e configure seu projeto para usar o Java 11.

Não pode haver equipes com o mesmo algoritmo.

Você pode testar sua classe com o método disponibilizado pelo professor no Github da turma.

Trabalho em duplas ou trios para entrega até 27/06 às 18h30.

Entregue apenas a classe (arquivo .java, sem compactar) e o link do vídeo no Youtube por meio da tarefa no AVA (Unidade 5).

Os nomes dos membros da equipe devem figurar como comentário no código-fonte e devem ser informados ao professor (<u>marcel@furb.br</u>) para constarem como grupo no AVA.

Prova 3 em 27/06.

FUNDAÇÃO UNIVERSIDADE REGIONAL DE BLUMENAU CENTRO DE CIÊNCIAS EXATAS E NATURAIS DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO Professor Marcel Hugo ALGORITMOS E ESTRUTURAS DE DADOS

Trabalho 4 2022/1

Equipes

Block Sort	
Bucket sort *	Pedro Schumann e Andrei Robson da Silva
Counting sort *	Arthur, Luiz Gustavo e Nadir
Cubesort	
Cycle sort	José João e Hiago
Gnome sort	Felipe F Weber, Eduarda, Marino
Heapsort	Ana Carolina da Silva, Gustavo Baroni Bruder, Felipe Melio Tomelin
Insertion sort	Rubens e Pedro Appel
Introsort	Michel e Guilherme
Merge sort	Nathan e Gabriel Toniolo
Quick sort	Bruna e Julia
Radix sort LSD *	Pedro Cintra e Kauê
Radix sort MSD *	Gabriel Adriano, Rafael, Leonardo S Nunes
Selection sort	Sarah
Shell sort	Bernardo e Vinicius
Timsort	
Tree Sort	Gabriel Arno, Lucas Mota e Victor

^{*} algoritmos de não comparação. Não irá seguir a interface Sort e o método sort deve ter a seguinte sintaxe: void sort(int[])