

Atividades iguais serão zeradas.

Exercícios que envolvam codificações iguais serão zerados.

1) Realize a simulação com o Algoritmo Merge Sort até que ocorra a ordenação completa do Vetor Abaixo:

Considere o vetor abaixo:

26	69	25	53	59	27	41	0	33	16	35	43
----	----	----	----	----	----	----	---	----	----	----	----

2) Realize a implementação do Algoritmo Merge Sort para o vetor acima, incluindo o método de Intercalação.

3) Realize a implementação do Algoritmo Merge Sort para uma Lista Duplamente Ligada considerando o seguinte cenário:

- Implemente a classe Aluno com os atributos ID, Nome completo, Curso e Semestre
- Implemente a classe Disciplina com os atributos ID, Nome da Disciplina
- Implemente a classe Média com os atributos ID do Aluno, ID da Disciplina e Média Final
- As classes Aluno, Disciplina e Média devem realizar as seguintes operações:
 - Inserir um elemento no final da Lista
 - Inserir um elemento no início da Lista
 - Remover um elemento do final da Lista
 - Remover um elemento do início da Lista
- Realize a ordenação dos Alunos pela média final utilizando o Algoritmo Merge Sort e apresente em console as seguintes informações:
 - Nome do Aluno
 - Disciplina Cursada
 - Média Final obtida
- Apresente a quantidade de alunos com média maior ou igual a 8.

4) Realize a ordenação do algoritmo Quick Sort até que ocorra a ordenação completa do vetor abaixo:

PIVÔ=99:

149	192	47	152	159	195	61	66	17	167	118	64	27	80	30	105
-----	-----	----	-----	-----	-----	----	----	----	-----	-----	----	----	----	----	-----

5) Realize a implementação do algoritmo Quick Sort para o vetor acima, incluindo o método de Particionamento.

6) Considerando o cenário descrito no Exercício 3, realize a implementação do Algoritmo Quick Sort para uma Lista Duplamente Ligada.

7) Simule todos os passos até o estado final da árvore binária para os seguintes elementos: 25, 5, 30, 8, 20, 31, 3, 99, 100, 200, 205, 198. Explique quais tipos de percursos poderiam ser usados para a árvore.