**Pesquisa e ordenação**

1) O que é ordenar?

Em Java, ordenar geralmente se refere ao processo de organizar os elementos de uma coleção, como um array ou uma lista, em uma ordem específica, como crescente ou decrescente.

2) Por que ordenar?  
  
Para facilitar a pesquisa, recuperação de dados, melhorar a legibilidade e a apresentação

3) Os algoritmos de ordenação são categorizados por:  
**- complexidade**  
**Complexidade** refere-se ao tempo e/ou espaço necessário para executar um algoritmo. Em ordenação, isso geralmente é medido em termos de complexidade de tempo, que indica quanto tempo o algoritmo levará para ordenar uma lista em função do número de elementos na lista.

**- estabilidade**A **estabilidade** de um algoritmo de ordenação se refere à preservação da ordem relativa dos elementos iguais na lista original. Em outras palavras, se dois elementos têm o mesmo valor e aparecem em uma certa ordem na lista original, um algoritmo estável garantirá que eles manterão essa ordem na lista ordenada.  
4) Qual ou quais os melhores métodos de ordenação?

**Quick Sort**: é um dos algoritmos de ordenação mais rápidos na prática devido ao seu bom desempenho médio, com uma complexidade de tempo O(n log n) na maioria das situações. No entanto, pode ter uma complexidade de O(n²) no pior caso, especialmente se o pivô escolhido não for ideal.

**Merge Sort:** é conhecido por sua eficiência e estabilidade. Com uma complexidade de tempo de O(n log n) em todos os casos, é uma escolha sólida quando você precisa de uma ordenação consistente e confiável, especialmente para grandes conjuntos de dados ou listas ligadas.

**Heap Sort:** oferece uma complexidade de tempo garantida de O(n log n) em todos os casos, o que o torna uma opção confiável quando a eficiência é crucial. Além disso, é um algoritmo in-place, usando O(1) de memória adicional. No entanto, não é estável e pode ser menos eficiente em prática comparado ao Quick Sort devido a maiores constantes e sobrecarga associada à manutenção do heap.

5) O que os métodos de ordenação bolha, seleção e inserção tem em comum os seguintes aspectos:

- Todos esses algoritmos possuem uma complexidade de tempo O(n²) no pior caso.  
- São bastante simples e intuitivos  
- Os três métodos de ordenação são in-place, o que significa que eles não requerem memória adicional significativa além da própria lista a ser ordenada.

- Todos esses algoritmos operam em um estilo de comparação e troca (ou inserção), onde elementos são comparados uns com os outros e trocados ou movidos para alcançar a ordem correta.

Artur Kieling