**Introdução**

Ordenação trata-se basicamente de colocar números, letras, códigos, ou qualquer tipo de informação em ordem. A mesma será definida previamente, podendo ser crescente, assim como na maioria dos casos, decrescente ou de outras formas. Entretanto existem outros tipos de ordenação não muito comuns, cujas quais há a necessidade da criação de um algoritmo afim de resolvê-las. Para isso utiliza-se linguagens de programação, onde os algoritmos são criados e testados. Podem ser elas: Java, C, C#, entre outras.

O motivo mais popular para se colocar os dados em ordem é o fato de podermos acessá-los com maior facilidade, sem haver a necessidade de ficar buscando um por um. Em busca dessa facilidade foram criados vários algoritmos de grande importância para o mundo da computação, os mais conhecidos são: *bubble sort (*ordenação por flutuação), *heap sort* (ordenação por *heap)*, *insertion sort* (ordenação por inserção), *merge sort* (ordenação por mistura) e *quick sort*. Existem outros diferentes algoritmos mais simples e/ou mais complexos que os citados, porém estes são os mais utilizados. Apesar disso, os mais utilizados por universidades, cursos, etc. para fins de estudo são: *bubble sort,*  s*election sort e quick sort,* muito por conta de serem de mais fácil aprendizagem.

*Bubble sort*: é tido como o algoritmo de ordenação mais simples entre os vários existentes. O que ele faz é basicamente comparar elementos adjacentes, como por exemplo: compara a primeira posição com a segunda, a segunda com a terceira e assim segue.

*Heap sort*: usa a estrutra *heap binário* para ordenar os dados conforme são inseridos no código. No final do mesmo os dados podem ser removidos de forma sucessiva da raíz *heap* da forma desejada.

*Insertion sort*: compara o novo elemento do vetor (normalmente chamado de chave) com cada um dos elementos à sua esquerda até que o mesmo seja colocado na posição correta.

*Merge sort*: utiliza o método “dividir para conquistar”, onde divide o vetor original em 2 sub-vetores e assim por diante até restar apenas um elemento em cada sub-vetor. Feito isso, junta os sub-vetores parte por parte e os ordena, até juntar o vetor completo.

*Quick sort*: algoritmo recursivo que também usa o método de dividir para conquistar. Escolhemos um elemento do vetor como pivô e a partir dele o algoritmo reorganiza os dados onde o vetor se divide em dois: à esquerda os valores menores que o pivô e à direita os valores maiores que o pivô. Repete o método ordenando de forma recursiva as duas sub-sequências e depois as une, formando assim o vetor ordenado.

<https://pt.wikiversity.org/wiki/Introdu%C3%A7%C3%A3o_%C3%A0s_Estruturas_de_Dados/Algoritmos_de_Ordena%C3%A7%C3%A3o> – Geral

<https://www.embarcados.com.br/algoritmos-de-ordenacao-bubble-sort/> - Bubble

<https://www.cos.ufrj.br/~rfarias/cos121/aula_09.html> - Heap

<https://pt.khanacademy.org/computing/computer-science/algorithms/insertion-sort/a/insertion-sort> - Insertion

<http://www.inf.ufsc.br/~r.mello/ine5384/18-OrdenacaoDados4.pdf> - Merge

<http://www.dcc.fc.up.pt/~pbv/aulas/progimp/teoricas/teorica19.html> - Quick