

# Introdução

## Bubble Sort

Bubble Sort é um algoritmo de ordenação que pode ser aplicado em Arrays e Listas dinâmicas. Se o objetivo é ordenar os valores em forma decrescente, então, a posição atual é comparada com a próxima posição e, se a posição atual for maior que a posição posterior, é realizada a troca dos valores nessa posição. Caso contrário, não é realizada a troca, apenas passa-se para o próximo par de comparações.

Se o objetivo é ordenar os valores em forma crescente, então, a posição atual é comparada com a próxima posição e, se a posição atual for menor que a posição posterior, é realizada a troca. Caso contrário, a troca não é feita e passa-se para o próximo par de comparação.

Um array ou lista pode estar já ordenado no momento em que se solicita a ordenação, dessa forma, esta situação tem de ser considerada na implementação do algoritmo.

## Selection Sort

Selection Sort sua ideia consiste em ordenar a lista “selecionando” a cada iteração o menores itens possíveis e os colocam da esquerda para a direita.

Assim como o Bubble Sort, é necessário para cada item da lista percorrê-la toda (logo, serão necessários dois loops: um para cada elemento da lista e outro para cada um desses elementos percorrer toda a lista).

Suas principais vantagens estão na fácil implementação do algoritmo, além de ocupar pouca memória se comparado a algoritmos como quick e merge sort que utilizam a estratégia de “dividir e conquistar”.

A desvantagem é que um dos algoritmos de maior tempo de execução em todos os casos, o que o torna uma opção ruim para listas com um grande número de itens.

## **insertion sort**

A ideia do insertion sort consiste em ordenar os itens, inserindo-os na posição corresponde da lista. Imagine que você esteja comprando uma carta e que você já tenha algumas cartas na sua mão, as quais você já fez o trabalho de ordenar corretamente. Você não sabe que carta virá, mas por motivos de organização para o jogo, gostaria de manter a ordenação conforme as cartas vão para sua mão. No momento em que você saca uma carta, o valor se torna a referência para você fazer a ordenação (chamaremos de “chave” daqui para a frente) e então, começa a comparar o valor com as cartas da sua mão (começando da última carta à direita) em direção à esquerda, até que encontre uma carta que seja maior ou igual à esta. Finalmente, você descobre que posição esta carta deve ficar de acordo com a ordem das cartas. Porém, as outras cartas que já estão na sua mão e são maiores precisam ser movidas para a direita para essa carta ser “encaixada” ou inserida naquela posição.

## **Quicksort**

O algoritmo quicksort é um método de ordenação muito rápido e eficiente, inventado por C.A.R. Hoare em 1960, quando visitou a Universidade de Moscovo como estudante. Naquela época, Hoare trabalhou em um projeto de tradução de máquina para o National Physical Laboratory. Ele criou o quicksort ao tentar traduzir um dicionário de inglês para russo, ordenando as palavras, tendo como objetivo reduzir o problema original em subproblemas que possam ser resolvidos mais fácil e rápido. Foi publicado em 1962 após uma série de refinamentos.

## **Mergesort**

Mergesort Sua ideia básica consiste em Dividir (o problema em vários subproblemas e resolver esses subproblemas através da recursividade) e Conquistar (após todos os subproblemas terem sido resolvidos ocorre a conquista que é a união das resoluções dos subproblemas). Como o algoritmo Merge Sort usa a recursividade, há um alto consumo de

memória e tempo de execução, tornando esta técnica não muito eficiente em alguns problemas. Tem como desvantagem a Utilização de funções recursivas; Gasto extra de memória.