

Bubble Sort:

Compara pares de elementos adjacentes
e troca de lugares se estiverem na ordem errada

esse processo se repete até que mais nenhuma troca seja necessária

Performance:

- . Melhor Caso: $O(N)$
 - . Pior Caso: $O(N^2)$
 - . Não recomendado para grande quantidade de conjunto de dados
-

Insertion Sort:

Similar a ordenação de cartas de baralho

Pega-se uma carta por vez e a coloca em sua devida posição na mão

esse processo se repete até que mais nenhuma troca seja necessária

Performance:

- . Melhor Caso: $O(N)$
 - . Pior Caso: $O(N^2)$
 - . Não recomendado para grande quantidade de conjunto de dados
 - . Estável , ou seja , não altera ordem de dados iguais
 - . Ordena em tempo real , conforme recebe os dados
-

Selection Sort:

A cada passo , procura o menor valor do array e o coloca na primeira posição

Descarta-se então a primeira posição do array e repete-se o processo para a segunda posição

Logo , vetor estará ordenado

esse processo se repete até que mais nenhuma troca seja necessária

Performance:

- . Melhor Caso: $O(N^2)$
 - . Pior Caso: $O(N^2)$
 - . Não recomendado para grande quantidade de conjunto de dados
 - . Estável , ou seja , não altera ordem de dados iguais
-

Merge Sort:

Ideia básica : Dividir e Conquistar

Divide de forma recursiva , o conjunto de dados até que cada subconjunto possua 1 elemento

Combina 2 subconjuntos de forma a obter 1 subconjunto MAIOR e ORDENADO

Esse processo se repete até que exista apenas um conjunto (ordenado)

Performance:

- . Melhor Caso: $O(N \log N)$

- . Pior Caso: $O(N \log N)$
- . Estável , ou seja , não altera ordem de dados iguais
- . Recursivo e usa vetor auxiliar durante a ordenação (Desvantagem)

Shell Sort:

Shell sort é o mais eficiente algoritmo de classificação dentre os de complexidade quadrática.

Derivado do Insection Sort

Faz trocas a uma certa distância (que diminui a cada passada)

Primeiro compara elementos separados por "h" posições e logo após os ordenada

Faz isto até que $h=1$

Perfomance:

- . Melhor Caso: $O(N \log^2 N)$
- . Pior Caso: $O(N \log^2 N)$
- . Possui excelentes desempenhos em N muito grandes
- . Não é estável