

Lógica e estratégia dos algoritmos de ordenação

1. Bubble Sort — Estratégia: Comparação Direta

O Bubble Sort é um algoritmo simples baseado em comparações diretas entre elementos adjacentes:

1.1 Lógica:

- Percorre o vetor várias vezes.
- A cada passagem, compara pares de elementos vizinhos. Se estiverem na ordem errada, eles são trocados.
- Após cada passagem, o maior elemento “borbulha” para o final do vetor.

1.2 Características:

- Fácil de entender.
 - Pode parar cedo caso o vetor já esteja ordenado.
-

2. Insertion Sort — Estratégia: Inserção Direta em Sublista Ordenada

O Insertion Sort funciona como organizar cartas na mão:

2.1 Lógica:

- Considera que a parte esquerda do vetor já está ordenada.
- Pega o próximo elemento e insere na posição correta nessa sublista ordenada.
Para inserir, desloca elementos maiores à direita.

2.2 Características:

- Muito eficiente em vetores quase ordenados.
-

3. Selection Sort — Estratégia: Seleção do Menor Elemento

O Selection Sort escolhe repetidamente o menor elemento da parte não ordenada:

3.1 Lógica:

- Para cada posição i :
 - Busca o menor elemento do intervalo $[i..n-1]$.
 - Troca esse menor com o elemento da posição i .
- Assim, após a i -ésima iteração, os i primeiros elementos já estão no lugar correto.

3.2 Características:

- Realiza sempre exatamente $n-1$ trocas, independentemente da ordem inicial.
-

4. Merge Sort — Estratégia: Divisão e Conquista

O Merge Sort usa a poderosa técnica de Dividir para Conquistar.

4.1 Lógica:

- Divide o vetor em duas metades.
- Ordena as duas metades recursivamente (divisão até vetores de 1 elemento).
- Intercala (merge) as duas metades ordenadas em um único vetor também ordenado.

4.2 Características:

- Exige memória auxiliar para intercalar as metades.

5. Quick Sort — Estratégia: Divisão e Conquista (Particionamento)

O Quick Sort também usa divisão e conquista, mas de forma diferente do Merge Sort:

5.1 Lógica:

- Escolhe um pivô.
- Reorganiza o vetor (partição):
 - Elementos \leq pivô ficam à esquerda.
 - Elementos $>$ pivô ficam à direita.
- Recursivamente ordena as duas partes.

5.2 Características:

- Ordena in-place (quase sem memória extra).

6. Shell Sort — Estratégia: Insertion Sort Generalizado com Intervalos

O Shell Sort melhora o Insertion Sort usando intervalos (gaps):

6.1 Lógica:

- Começa comparando e inserindo elementos distantes (gap grande).
- Reduz o gap progressivamente.
- Quando o gap chega a 1, o algoritmo vira um insertion sort final — mas o vetor já está quase ordenado.

6.2 Características:

- Desempenho depende da sequência dos gaps.
- Boa performance prática para vetores médios e grandes.

7. Resumo Geral das Estratégias

Algoritmo	Estratégia Principal	Tipo de Operação Central
Bubble Sort	Comparação direta	Troca de elementos adjacentes
Insertion Sort	Inserção por deslocamento	Inserção em sublista ordenada
Selection Sort	Seleção do menor elemento	Troca do menor com posição atual
Merge Sort	Divisão e Conquista	Intercala vetores ordenados

Quick Sort	Divisão e Conquista (partição)	Rearranjo baseado em pivô
Shell Sort	Inserção com gaps	Insertion sort modificado