MÉTODOS DE ORDENAÇÃO E ORGANIZAÇÃO DE ARQUIVOS

# MÉTODOS CLÁSSICOS DE ORDENAÇÃO

Os algoritmos de classificação são formas de organizar uma matriz de itens do menor para o maior valor. Esses algoritmos podem ser utilizados para organizar dados confusos, tornando-os mais fáceis de usar.

Os primeiros algoritmos de Inserção Direta e de Ordenação por Shell classificam arrays com inserção, que é quando os elementos são inseridos no lugar certo. Os outros dois, Ordenação por Bolha e Ordenação Rápida, classificam arrays com troca, que é quando os elementos se movem ao redor do array. O último, Ordenação por Heap, classifica através da seleção, onde os elementos corretos são selecionados à medida que o algoritmo percorre o array.

O Big O é usado para denotar a complexidade de tempo de um algoritmo ou espaço que ele ocupa. A maneira como as pessoas podem calcular isso é identificando o pior caso para o algoritmo alvo e formulando uma função de seu desempenho dada a quantidade n de elementos.

Por exemplo, se houvesse um algoritmo que procurasse o número 2 em uma matriz, o pior caso seria se o 2 estivesse no final da matriz. Portanto, a notação Big O seria O(n), pois teria que percorrer toda a matriz de n elementos antes de encontrar o número 2.

Assim, a **Ordenação por Inserção Direta** é um dos algoritmos de ordenação mais básicos que, essencialmente, insere um elemento na posição correta de uma lista já ordenada. Ele, geralmente, é adicionado no final de um novo array e se move para baixo até encontrar um elemento menor que ele mesmo (a posição desejada).

A **Ordenação por Shell** é uma classificação por inserção que primeiro classifica parcialmente seus dados e, em seguida, termina a classificação executando um algoritmo de classificação por inserção em todo o array. Geralmente, ele começa escolhendo pequenos subconjuntos e classificando os arrays.

A **Ordenação por Bolha** compara elementos adjacentes de uma matriz e organiza esses elementos. Seu nome vem do fato de que grandes números tendem a “flutuar” (bolha) para o topo. Ele percorre uma matriz e verifica se o número em uma posição é maior que o número na posição seguinte, o que resultaria no aumento do número.

A **Ordenação Rápida** é um dos algoritmos de ordenação mais eficientes e isso faz dele um dos mais usados também. A primeira coisa a fazer é selecionar um número pivô, este número irá separar os dados. Assim, à esquerda ficarão os números menores que ele e os números maiores ficarão à direita.

A **Ordenação por Heap** na primeira etapa do algoritmo, é criada uma árvore com os valores a serem ordenados, começando pela esquerda, criamos o nó raiz, com o primeiro valor. Então, um nó filho esquerdo é criado e o próximo valor é inserido. Neste momento, avaliamos, se o valor definido para o nó filho é maior que o valor no nó raiz, alteramos os valores. Isso é feito com toda a árvore. A ideia inicial é que os nós pais sempre tenham valores maiores que os nós filhos. Assim, ao final da primeira etapa, criamos um vetor começando com o valor da raiz e caminhando da esquerda para a direita preenchendo o vetor. Então, começamos a comparar os valores dos nós pai e filho procurando o maior valor entre eles e, quando o encontramos, trocamos de lugar reordenando os valores.

# **ANÁLISE DA COMPLEXIDADE DOS MÉTODOS DE ORDENAÇÃO**

A complexidade não tem nenhuma definição formal, serve apenas para definir a taxa de eficiência na qual uma tarefa é executada. Em estruturas de dados e algoritmos, existem dois tipos de complexidades que determinam a eficiência de um algoritmo, elas são:

* Complexidade do espaço: é a memória total consumida pelo programa para sua execução;
* Complexidade de tempo: é definida como os tempos em que se espera que a instrução numérica, em particular, seja executada, em vez do tempo total que é gasto.

Na ciência da computação, a complexidade de tempo de um algoritmo é expressa em notação O grande. Verifiquemos, a seguir, algumas complexidades:

* O **O(1)** denota o tempo constante;
* **O(log n)** significa diminuir a cada instância para as operações;
* O **O(n)** denota tempo linear. O(n) significa que o desempenho é diretamente proporcional ao tamanho da entrada;
* O **O(n^2)** denota tempo quadrático. O(n^2) significa que o desempenho é diretamente proporcional ao quadrado da entrada tomada.

O melhor caso é quando a lista de elementos fornecida já está classificada. É por isso que a Ordenação por Bolha não é considerada boa o suficiente quando o tamanho da entrada é muito grande.

**Bubble sort** é um algoritmo de ordenação simples, onde os elementos são ordenados comparando cada par de elementos e trocando-os, se um elemento não seguir a ordem de ordenação desejada.

* Complexidade média do tempo do caso: O(n^2);
* Complexidade de tempo do pior caso: O(n^2);
* Complexidade de tempo do melhor caso: O(n).

Na **Ordenação por Seleção**, pegamos um elemento e passamos para sua posição correta. Esse processo é realizado desde que todos eles sejam classificados na ordem desejada.

* Complexidade média do tempo do caso: O(n^2);
* Complexidade de tempo do pior caso: O(n^2);
* Complexidade de tempo do melhor caso: O(n^2).

A **Ordenação por Inserção Direta** funciona no fenômeno pegando entradas e colocando-as na ordem ou local corretos.

* Complexidade de tempo do melhor caso: O(n);
* Complexidade de tempo média e de pior caso: O(n^2).

Já a **Ordenação Rápida** grandes matrizes de entrada são divididas em submatrizes menores, que são classificadas recursivamente e mescladas em uma enorme matriz após a classificação.

* Melhor e média complexidade de tempo: O(n log n);
* Complexidade de tempo do pior caso: (n^2).

Na **Ordenação por Mistura** a matriz de entrada é dividida ao meio e, em seguida, essas metades são classificadas. Após a classificação, as duas submatrizes divididas ao meio são mescladas em uma para formar uma matriz classificada completa.

* Melhor e média complexidade de tempo: O(n log n)
* Complexidade de tempo do pior caso: O(n log n)

# **ARQUIVOS DE ACESSO SEQUENCIAL**

O tipo mais comum é o **arquivo de acesso sequencial**, que possuem uma organização simples, se modificam facilmente com editores de texto e são úteis para exportar dados entre programas. Um arquivo sequencial contém registros organizados pela ordem em que foram inseridos, que é fixa. Assim, registros em arquivos sequenciais podem ser lidos ou escritos apenas sequencialmente. Depois de colocar um registro em um arquivo sequencial, não podemos encurtar, aumentar ou excluir o registro. No entanto, podemos reescrever um registro se o comprimento não mudar. Novos registros são adicionados ao final do arquivo.

# **ARQUIVOS DE ACESSO DIRETO**

A **organização direta de arquivos** permite que os dados sejam recuperados rapidamente de maneira aleatória, independentemente da forma como os dados foram originalmente armazenados. Nesse método, a chave exclusiva usada para organizar arquivos (como um número de CPF) é convertida diretamente em um endereço de memória, usando uma fórmula matemática chamada algoritmo de hash.

O **endereço** é um número de cinco a sete dígitos relacionado às características físicas do meio de armazenamento. Quando um arquivo é criado, esse endereço determina onde o registro é gravado. Durante a recuperação, ele determina onde localizar o registro.

O design de arquivo de acesso direto é muito mais eficiente do que pesquisar um arquivo de dados inteiro para um registro específico. É útil quando as informações devem ser atualizadas e recuperadas rapidamente e quando as informações atuais são cruciais.

As vantagens do processamento de acesso direto e do design de arquivos são as seguintes:

* Os dados de transação podem ser usados diretamente para atualizar os registros mestre por meio de terminais online sem primeiro serem ordenados. As transações são processadas à medida que ocorrem;
* O arquivo mestre não é lido completamente toda vez que ocorre a atualização, somente os registros mestre a serem atualizados são acessados, isso economiza tempo e dinheiro;
* Obter acesso a qualquer registro em um arquivo de acesso direto leva apenas uma fração de segundo;
* Arquivos de acesso direto oferecem flexibilidade no tratamento de consultas;
* Vários arquivos podem ser atualizados ao mesmo tempo pelo uso de processamento de acesso direto.

As desvantagens do projeto de acesso direto incluem:

* Durante o processamento, o registro original é substituído pelo registro atualizado. Com efeito, é destruído;
* Como muitos usuários podem ter acesso a registros armazenados em dispositivos de acesso direto em sistemas online, as chances de destruição acidental de dados são maiores do que com processamento sequencia;
* O acesso direto pode apresentar problemas de segurança para as organizações. Os usuários podem acessar informações confidenciais;
* A implementação de sistemas de acesso direto é muitas vezes difícil devido à complexidade e ao alto nível de suporte de programação (software) que esses sistemas precisam.

# **ARQUIVOS DE ACESSO INDEXADO**

Um arquivo de computador de acesso indexado é referido como um arquivo indexado porque permite acesso aleatório rápido a qualquer registro com base em sua chave de arquivo.

Com acesso sequencial, o processamento iniciaria com o registro e prosseguiria sequencialmente pelo arquivo até encontrar o registro que contém as informações necessárias. Com acesso aleatório, o processamento aproveita uma estrutura de índice e recupera o registro que contém as informações necessárias.

Assim, os arquivos indexados são criados com um ou mais índices especificados pelo programador. O método indexado mais comum é o VSAM. No VSAM, o arquivo indexado possui um índice principal que geralmente é o número de identificação e índices alternativos opcionais. Por exemplo, em um arquivo pessoal, o número do seguro social seria frequentemente o índice principal e o nome do funcionário o índice alternativo. Se não houver tal número, será solicitado o nome da pessoa. Isso significa que, a primeira tentativa de localizar o registro no computador seria a chave principal, o número do CPF ou RG, caso não esteja disponível, o nome serve como chave alternativa.

É preciso lembrarmos que o método de acesso se refere a como os arquivos no disco serão acessados. Arquivos com organização sequencial só podem ser acessados sequencialmente. Já os arquivos com organização indexada podem ter um modo de acesso sequencial, aleatório ou dinâmico.

O acesso sequencial significa que os registros só podem ser lidos em sequência, porém com a organização indexada o ponto de partida não precisa estar no início do arquivo. O índice pode ser usado para iniciar o processamento sequencial em qualquer ponto do arquivo.