

---

---

# Busca Sequencial e Binária

Aluno: Abrahão Picanço Neres

---

# Busca Sequencial

A Busca Sequencial, também conhecida como Busca Linear, é um algoritmo simples utilizado para encontrar um valor dentro de uma lista (ou array). Ele percorre a lista de elementos, verificando cada item sequencialmente até encontrar o valor desejado ou até o final da lista.



# Algoritmo

O algoritmo de **Busca Sequencial** pode ser descrito da seguinte maneira:

## → Inicialização

Inicie a busca a partir do primeiro elemento do array.

## → Comparação

Compare o valor do elemento atual com o valor que está sendo procurado.

## → Condição de Encontro

Se os valores forem iguais, retorne o índice correspondente ao elemento encontrado.

## → Iteração

Caso contrário, avance para o próximo elemento do array e repita o processo de comparação.

## → Condição de Término

Se o valor não for encontrado após a verificação de todos os elementos, retorne um valor que indique a ausência do item (comumente representado por -1).

# CÓDIGO

```
// Função de Busca Sequencial
int buscaSequencial(int array[], int tamanho, int valorProcurado) {
    // Percorre o array elemento por elemento
    for (int i = 0; i < tamanho; i++) {
        // Se o elemento atual for igual ao valor procurado
        if (array[i] == valorProcurado) {
            return i; // Retorna o índice onde o valor foi encontrado
        }
    }
    // Se o valor não for encontrado, retorna -1
    return -1;
}
```



# Vantagens

1. O algoritmo de busca sequencial é de fácil compreensão e implementação, sendo uma solução direta para o problema de localização de elementos em uma lista.
2. Diferente de outros métodos de busca, como a busca binária, a busca sequencial não requer que a lista esteja previamente ordenada, o que a torna aplicável a uma ampla variedade de cenários.
3. Em casos onde o volume de dados é reduzido, a busca sequencial pode ser uma opção eficiente, uma vez que o tempo de execução é aceitável para listas de tamanho limitado.

# Desvantagens

1. O algoritmo possui complexidade temporal  $O(n)$ , o que significa que, no pior caso, ele precisa percorrer todos os elementos da lista para encontrar o valor desejado. Isso o torna ineficiente em termos de desempenho para listas grandes.
2. Em cenários com listas de tamanho considerável, a busca sequencial tende a ser lenta, uma vez que o tempo de execução aumenta linearmente com o crescimento do número de elementos. Isso a torna inadequada para aplicações que exigem alta eficiência em grandes conjuntos de dados.

# Busca Binária

A Busca Binária é um algoritmo de busca eficiente que funciona em listas ordenadas. Ao invés de verificar cada elemento, como na busca sequencial, a busca binária divide a lista ao meio, verificando se o valor procurado está na metade inferior ou superior. Esse processo é repetido até encontrar o valor ou determinar que ele não está presente

# CÓDIGO

```
// Função de Busca Binária
int buscaBinaria(int array[], int tamanho, int valorProcurado) {
    int baixo = 0;
    int alto = tamanho - 1;

    while (baixo <= alto) {
        int meio = baixo + (alto - baixo) / 2; // Calcula o índice médio

        // Verifica se o valor está no meio
        if (array[meio] == valorProcurado) {
            return meio; // Retorna o índice se encontrado
        }

        // Se o valor for menor, ignora a metade superior
        if (array[meio] > valorProcurado) {
            alto = meio - 1;
        }

        // Se o valor for maior, ignora a metade inferior
        else {
            baixo = meio + 1;
        }
    }

    // Se o valor não for encontrado, retorna -1
    return -1;
}
```



# Vantagens

1. A complexidade do algoritmo, tornando-o muito mais rápido em listas grandes quando comparado à busca sequencial.
2. A cada iteração, a lista é dividida pela metade, o que reduz o número de elementos a serem verificados.

# Desvantagens

1. Para que a busca binária funcione, a lista precisa estar ordenada.
2. Em comparação com a busca sequencial, a implementação da busca binária é mais difícil de entender e implementar corretamente.