



ED1 01: Introdução à Complexidade e Análise Assintótica.

Fabio Irigon Pereira

O que são Estruturas de Dados?



- Conjunto de técnicas para organizar e armazenar dados de forma eficiente.
- Diferentes estruturas oferecem vantagens para diferentes tipos de problemas.
- Exemplo: Listas, Pilhas, Filas, Árvores, Grafos.

Motivação e Aplicações



- Eficiência computacional em algoritmos.
- Exemplos:
 - Recebimento e processamento de pacotes de dados.
 - Algoritmos de decisão.
 - Algoritmos de roteamento.

Complexidade de tempo e espaço

O que é Complexidade?



- Medida da eficiência de um algoritmo considerando **tempo de execução** e **uso de memória**.
- Independente do hardware específico, foca no comportamento para entradas grandes.

Complexidade de Tempo



- Mede o crescimento do número de operações que um algoritmo executa conforme o tamanho da entrada n cresce.
- Exemplos:
 - $O(1)$ – Constante: Acesso direto a um elemento em um array.
 - $O(n)$ – Linear: Percorrer um array de tamanho n .
 - $O(n^2)$ – Quadrático: Algoritmo de ordenação por seleção.
 - $O(\log n)$ – Logarítmico: Busca binária.
 - $O(2^n)$ – Exponencial: Algoritmos de força bruta em problemas de otimização
 - (chutar password).

Complexidade de Espaço



- Mede a quantidade de memória usada pelo algoritmo.
- Inclui:
 - Memória fixa: Variáveis globais, constantes.
 - Memória dinâmica: Alocação durante execução (malloc em C).

Análise Assintótica: O , Ω e Θ



A notação assintótica descreve o comportamento de um algoritmo conforme o tamanho da entrada cresce para infinito.

- O (Big-O) \rightarrow Limite superior: pior caso.
 - Exemplo: Ordenação por inserção $\rightarrow O(n^2)$
- Ω (Big-Omega) \rightarrow Limite inferior: melhor caso.
 - Exemplo: Busca sequencial em uma lista ordenada pode ser $\Omega(1)$ no melhor caso.
- Θ (Theta) \rightarrow Limite exato: comportamento médio.
 - Exemplo: Busca binária $\rightarrow \Theta(\log n)$ em todos os casos.


Exemplo:



Busca do maior elemento.

Algoritmo bubble sort.

Busca um elemento em árvore binária de busca.



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 40000

int main() {

    int arr[SIZE];
    srand(time(NULL));
    for (int i = 0; i < SIZE; i++)
        arr[i] = rand() % 10000;

    // Medindo tempo da busca do maior elemento
    clock_t start = clock();
    int maxElement = findMax(arr, SIZE);
    clock_t end = clock();
    printf("Maior elemento: %d, Tempo: %lf segundos\n", maxElement, ((double)(end - start)) / CLOCKS_PER_SEC);
}
```