

# Lista Teórica

## Estrutura de dados

### 2024.1

**Problema 1** Identifique a complexidade, em notação  $O$  (O-grande), das seguintes funções:

(a)  $100n + 0.01n^2$

Resposta:  $O(n^2)$

(b)  $k^2 + 5n^2$

Resposta:  $O(k^2 + n^2)$

(c)  $5 + 0.0001n^3 + 0.25n$

(d)  $0.3n + 5n^{1.5} + 2.5n^{1.75}$

(e)  $25 + 13$

(f)  $20m + 10n$

(g)  $10n + \log n$

**Problema 2** Preencha a tabela a seguir, informando o tempo gasto por um programa para resolver um problema com instâncias de tamanho  $n$ . Considere o tempo para cada instrução como 1 milésimo de segundo. A função de complexidade dá o número de passos para uma entrada de tamanho  $n$ .

	Tamanho da instância (n)				
Função de complexidade	1	2	3	4	5
$n + 10$					
$n^2 + n$					
$2^n$					

**Problema 3** Dado os seguintes algoritmos que calculam o somatório de todos os números inteiros entre 1 e  $n$ . Determine a complexidade de cada um utilizando a notação  $O$ :

(a) .

```
// retorna a soma 1 + 2 + 3 + ... + n
int sum(int n){
    int s = 0;
    for(int i = 1; i <= n; i++)
        s += i;
    return s;
}
```

(b) .

```
// retorna a soma 1 + 2 + 3 + ... + n
int sum(int n){
    return (n*(n+1))/2;
}
```

**Problema 4** O "melhor caso" de um algoritmo refere-se ao cenário em que o algoritmo executa sua tarefa no menor tempo possível ou utiliza a menor quantidade de recursos. Enquanto o "pior caso" refere-se ao maior gasto de tempo ou maior gasto de recursos.

Dado o seguinte algoritmo calcula se um inteiro  $n$  é primo ou não, determine:

```
// retorna true se n é primo, retorna false caso contrário
bool isPrime(int n){
    if(n == 1) return false;
    if(n == 2) return true;

    for(int i = 2; i <= n-1; i++)
        if(n % i == 0)
            return false;

    return true;
}
```

(a) Qual é a complexidade de tempo no melhor caso? Explique o caso.

(b) Qual é a complexidade de tempo no pior caso? Explique o caso.

(c) Qual é a complexidade de espaço no melhor caso?

(d) Qual é a complexidade de espaço no pior caso?

**Problema 5** Dado o seguinte algoritmo calcula se um inteiro  $n$  é primo ou não, determine:

```
// retorna true se n é primo, retorna false caso contrário
bool isPrime(int n){
    if(n == 1) return false;
    if(n == 2) return true;

    for(int i = 2; i*i <= n; i++)
        if(n % i == 0)
            return false;

    return true;
}
```

- (a) Qual é a complexidade de tempo no melhor caso?
- (b) Qual é a complexidade de tempo no pior caso?

**Problema 6** Dado o seguinte algoritmo que calcula o somatório de todos dígitos de um número inteiro  $n$ . Determine:

```
// retorna a soma dos digitos de um numero n
int digitSum(int n){
    int s = 0;
    while(n > 0){
        s += n%10;
        n = n/10;
    }
    return s;
}
```

- (a) Sua complexidade de tempo.
- (b) Sua complexidade de espaço.

**Problema 7** Cite uma vantagem e uma desvantagens de se usar listas sequenciais estáticas (arrays).

**Problema 8** Cite uma vantagem e uma desvantagem de se usar listas encadeadas.

**Problema 9** Cite uma vantagem e uma desvantagem de se usar listas simplesmente encadeadas sobre listas duplamente encadeadas.

**Problema 10** Preencha a seguinte tabela com a complexidade de pior caso para cada combinação de algoritmo e estrutura de dados:

Algoritmo	Estrutura		
	Lista sequencial	Lista simplesmente encadeada (com ponteiro para nó cauda)	Lista duplamente encadeada (com ponteiro para nó cauda)
Acessar o elemento na posição k	$O(k)$	$O(k)$	$O(k)$
Buscar elemento pelo valor			
Calcular a soma de todos os valores			
Inserir no começo			
Inserir no final			
Remover primeiro elemento			
Remover último elemento			
Remover elemento na posição k			

**Problema 11** [FUNDEP - 2012] Assinale a opção correta:

Os **dados** contidos em uma lista encadeada estão:

- (a) Ordenados sequencialmente
- (b) Sem ordem lógica ou física alguma.
- (c) Em ordem física e não, necessariamente, em ordem lógica.
- (d) Em ordem lógica e, necessariamente, em ordem física.
- (e) Em ordem lógica e não, necessariamente, em ordem física.