



**INSTITUTO FEDERAL**

Catarinense  
Campus Blumenau



Bacharelado em  
**CIÊNCIA DA  
COMPUTAÇÃO**

## Estruturas de Dados 1 - ED1

### Avaliação - Prova 1 - 2024.1

#### Identificação

Nome: Giovani Zanella da Maia

E-mail: Giovani.ZanelladaMaia123@gmail.com

Data: 05/07

Folhas: 12

Nota: **0.1**

#### Regras - Leia atentamente este quadro

- A prova é individual, sem consulta e não é permitido utilizar equipamentos eletrônicos.
- A prova será digitalizada. Não serão aceitas folhas de respostas com rasura, dobras, cliques, grampos, cola, corretivos, tintas ou qualquer coisa semelhante. Considere borda de 2cm em todos lados das páginas.
- As questões devem ser respondidas nos espaços reservados para as respostas, não serão aceitas folhas que não façam parte da prova. **Utilize caneta nas cores preta ou azul, também é possível utilizar lápis com graduação acima de 5B.** Trechos ilegíveis serão ignorados.
- Todas as questões devem ser respondidas com justificativas e apresentar o desenvolvimento completo do raciocínio da justificativa. Respostas vagas, justificativas nebulosas ou que configurem fuga do tema cancelarão o ponto da questão.
- O tempo de permanência mínima em sala é de 50 minutos. O atraso máximo para entrada na sala de prova é de 50 minutos.
- Todas as PÁGINAS devem possuir nome e serem enumeradas conforme [número da página]/[total de páginas]. Ao final da prova, coloque o número de folhas no espaço reservado na seção identificação.
- Plágio, consulta e conversas invalidarão individualmente a avaliação, será atribuída nota zero e o caso será informado à coordenação, isso poderá implicar em penalizações previstas no regime disciplinar discente.
- A soma dos pontos obtidos nas respostas será dividida pelo número de pontos totais e multiplicada por dez para gerar a nota da prova.
- Não utilize o verso das folhas.

# 1 Questões

1. É necessário implementar um programa para filtragem de dados. Uma parte desse problema é calcular a média dos valores que serão inseridos dinamicamente no programa. Os dados devem ser organizados em uma lista simplesmente encadeada e deve ser possível fazer o cálculo rápido da nova média em tempo  $\Theta[1]$  ao se inserir um novo elemento na lista. (ou seja, não é possível recalcular a média de todos os valores previamente armazenados).

1/10

- (a) (1 ponto) Faça uma estrutura para representar uma lista simplesmente encadeada e uma estrutura para representar o nó da lista e o algoritmo de inicialização dessa lista. (Ignore o problema do cálculo da média nesse instante).

```
class No {
    Int val;
    Int * proximo;
}
```

deveria ser pointer para No  
por que a lista possui um inteiro chamado proximo???

```
class Lista {
    No * cabeca;
    Int proximo;
}
```

```
Insere(No* no, Lista list)
// verificar se não tem no
se (list.size() == 0) {
    list.cabeca = no;
}
```

e foi feito no.prox = null?

```
else {
    // percorrer lista
    // Se tem no
    for (int i = 0; i < list.size(); i++) {
        se (list[i].proximo == null) {
            list[i].proximo = no;
        }
    }
}
```

```
main {
    No no = novo No(0, null);
    No na = novo No(1, null);
}
```

```
Lista list;
list.Insere(no, list);
```

Era para fazer a inicialização da lista e não inserção

list[i].proximo = 0



- (b) (2 pontos) Faça um algoritmo para inserção na lista simplesmente ligada. (Ignore o problema do cálculo da média nesse instante).

- (c) (2 pontos) Considerando uma sequência de dados  $(a_i)_n$  contendo  $n$  dados, a média aritmética é definida por:

$$\mu_n = \frac{1}{n} \sum_{i=1}^n a_i$$

Elabore uma equação para fazer o cálculo da média  $\mu_{n+1}$  que funcione em tempo  $\theta[1]$ . Dica: escreva as equações para  $\mu_n$  e  $\mu_{n+1}$ , considere que a média  $\mu_n$  já está calculada para gerar a média  $\mu_{n+1}$ .



É preciso ter um pouco de paciência para resolver as questões

$$\mu_n = \frac{1}{n} (a_1 + a_2 + \dots + a_n) \Leftrightarrow$$

$$n\mu_n = a_1 + a_2 + \dots + a_n$$

$$\mu_{n+1} = \frac{1}{n+1} (\underbrace{a_1 + a_2 + \dots + a_n}_{n\mu_n} + a_{n+1}) = \frac{1}{n+1} (n\mu_n + a_{n+1})$$

- (d) (3 pontos) Demonstre a corretude da equação obtida no item anterior utilizando indução matemática.

Trivial deixo a interpretação para o professor.

$$H[n] := \mu_{n+1} = \frac{1}{n+1} (a_1 + a_2 + \dots + a_n + a_{n+1}) = \frac{1}{n+1} (n\mu_n + a_{n+1})$$

Estou afirmando que a média de  $\mu_{n+1}$  pode ser calculada utilizando  $\mu_n$ , essa é a hipótese indutiva

Base da indução: aqui é necessário uma adaptação pois a média de 1 único elemento considerado a média anterior deve considerar  $\mu_0 = 0$

$$H_1[1]: \frac{1}{1} (a_1) = \frac{1}{1} (0 + a_1) = \frac{1}{1} (0 \cdot \mu_0 + a_1)$$

Indução  $n = k+1$

$$\begin{aligned} H[k]: \mu_{(k+1)+1} &= \frac{1}{(k+1)+1} (a_1 + a_2 + \dots + a_k + a_{k+1} + a_{k+2}) \\ &= \frac{1}{k+2} \left( (a_1 + a_2 + \dots + a_{k+1}) \frac{(k+1)}{(k+1)} + a_{k+2} \right) \\ &\stackrel{H}{=} \frac{1}{k+2} \left( \mu_{k+1} (k+1) + a_{k+2} \right) \end{aligned}$$

- [illegible]

- Blank lined paper for writing.



- 
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]



- [illegible]



4. Considere as seguintes definições envolvendo crescimento de funções:

**Definição 1**  $\Omega[g[n]] = \{f[n] : \text{existe uma constante real } c_1 \geq 0 \text{ e um natural } n_0 \text{ tal que } 0 \leq c_1 g[n] \leq f[n] \text{ para todo } n \geq n_0\}$ .

**Definição 2**  $O[g[n]] = \{f[n] : \text{existe uma constante real } c_2 \geq 0 \text{ e um natural } n_0 \text{ tal que } 0 \leq f[n] \leq c_2 g[n] \text{ para todo } n \geq n_0\}$ .

**Definição 3**  $\Theta[g[n]] = \{f[n] : \text{existem constantes reais } c_1, c_2 \geq 0 \text{ e um natural } n_0 \text{ tal que } 0 \leq c_1 g[n] \leq f[n] \leq c_2 g[n] \text{ para todo } n \geq n_0\}$ .

(a) (3 pontos) Demonstre a definição  $\Theta$  respeita a propriedade transitiva, ou seja,

$$f[n] \in \Theta[g[n]] \wedge g[n] \in \Theta[h[n]] \implies f[n] \in \Theta[h[n]]$$

## Referências

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

```

class No() {
    int valor;
    int * proximo;
}

```

```

}

```



```

criar no(valor) {
    //
}

```

```

class Lista() {
    No * cabeca;
    No * prox;
}

```

```

criar novo no {
    criar(*no) {
    this.cabeca = *no;
    //
}

```

```

iniciar(No* no, Lista list) {
    se (list.size() == 0) {
        // lista n foi iniciada
        list.cabeca = no;
    }
}

```

```

} else {
    while (no != null) {
    while (list.size() > 0) {
        for (int i = 0; i < list.size(); i++) {
            se list[i]
        }
    }
}

```

```

main() {
}

```

```

Lista list;

```

```

list.criar(80);

```

```

No no = novo NO(10, null);

```

```

NO

```