

## Estrutura de Dados e Análise de Algoritmos

Luca Cariolin - 119120572

**Valor:** 5 pontos

1. Analise o algoritmo apresentado a seguir e defina, linha a linha, qual a quantidade de vezes que o código é executado,  $T(n)$ , no pior caso, melhor caso e no caso médio:

	Melhor caso	Caso Médio	Pior Caso
<b>inteiro</b> i, j, k, x,			
A[n]; <b>início</b>	O(1)	O(1)	O(1)
A = [ ... ]	O(1)	O(1)	O(1)
<b>para</b> i de 1 até n <b>faça</b>	O(1)	O(N)	O(N)
k = i;	O(1)	O(1)	O(1)
x = A[i];	O(1)	O(1)	O(1)
<b>para</b> j de 1 até n <b>faça</b>	O(1)	O(N)	O(N)
<b>se</b> (A[j] < x) <b>então</b>	O(1)	O(1)	O(1)
k = j;	O(1)	O(1)	O(1)
x = A[k];	O(1)	O(1)	O(1)
<b>fim se</b>	O(1)	O(1)	O(1)
<b>fim para</b>			
A[k] =			
a[i]; A[i]			
= x;			
<b>fim para</b>			
<b>fim</b>			

## Estrutura de Dados e Análise de Algoritmos

2. Dada a estrutura

abaixo: `public class`

```
Aluno {  
    String  
    nome; int  
    matricula;  
    String  
    curso;  
}
```

Crie uma classe para armazenamento dos dados de vários alunos. As exigências são:

A) Utilize uma estrutura de alocação estática (vetor) para a tarefa

Implementado no código.

B) Crie um método para realizar a inserção de um novo aluno na primeira posição disponível no vetor

Implementado no código.

C) Crie um método para realizar a remoção de um aluno com base em um número de matrícula informado

Implementado no código.

D) Crie um método para informar a quantidade de posições vazias (sem alunos) na estrutura

Implementado no código.

E) Implemente a função de inserção que aumente de tamanho para caber mais posições. Toda vez que a estrutura estiver cheia ela deve aumentar em 5 posições.

Implementado no código.

3. Utilize a estrutura Aluno da questão anterior e faça:

A) Crie uma estrutura de alocação dinâmica na forma de uma Lista Simples Encadeada. O que deve mudar na classe Aluno?

Não há necessidade de modificar a classe aluno.

B) Crie um método para inserção de um novo aluno na última posição da estrutura

Implementado no código.

C) Crie um método para inserção de um novo aluno na primeira posição da estrutura

## Estrutura de Dados e Análise de Algoritmos

Implementado no código.

- D) Crie um método para inserção de um novo aluno em uma posição informada pelo usuário

Implementado no código.

- E) Crie um método que retorne a quantidade de alunos armazenados na lista

Implementado no código.

- F) Crie um método para realizar a remoção de um aluno com base em um número de matrícula informado

Implementado no código.

4. Avalie cada um dos métodos feitos nas questões 2 e 3 a respeito da quantidade de operações realizadas  $T(n)$  e diga:

- a) Qual o melhor caso?

### **Estrutura de alocação estática (vetor):**

- **Inserir Aluno:**  $O(1)$
- **Remover Aluno:**  $O(1)$
- **Quantidade de Posições Vazias:**  $O(1)$
- **Aumentar tamanho em 5 posições:**  $O(1)$

### **Lista Simples Encadeada:**

- **Inserir Aluno na última posição:**  $O(1)$
- **Inserir Aluno na primeira posição:**  $O(1)$
- **Inserir Aluno na posição informada usuário:**  $O(1)$
- **Quantidade de alunos Armazenados:**  $O(1)$
- **Remover aluno com matrícula:**  $O(1)$

- b) Qual o pior caso?

### **Estrutura de alocação estática (vetor):**

- **Inserir Aluno:**  $O(1)$
- **Remover Aluno:**  $O(N) + O(N)$
- **Quantidade de Posições Vazias:**  $O(N)$

- **Aumentar tamanho em 5 posições:**  $O(N)$

**Lista Simples Encadeada:**

- **Inserir Aluno na última posição:**  $O(N)$
- **Inserir Aluno na primeira posição:**  $O(1)$
- **Inserir Aluno na posição informada usuário:**  $O(N)$
- **Quantidade de alunos Armazenados:**  $O(N)$
- **Remover aluno com matrícula:**  $O(N)$

c) Qual o caso médio?

**Estrutura de alocação estática (vetor):**

- **Inserir Aluno:**  $O(1)$
- **Remover Aluno:**  $O(N) + O(N)$
- **Quantidade de Posições Vazias:**  $O(N)$
- **Aumentar tamanho em 5 posições:**  $O(N)$

**Lista Simples Encadeada:**

- **Inserir Aluno na última posição:**  $O(N)$
- **Inserir Aluno na primeira posição:**  $O(1)$
- **Inserir Aluno na posição informada usuário:**  $O(N)$
- **Quantidade de alunos Armazenados:**  $O(N)$
- **Remover aluno com matrícula:**  $O(N)$