



Centro Técnico Profissional UNIPAC
Fundação Presidente Antônio Carlos
AVALIAÇÃO DA DISCIPLINA
TÓPICOS INOVADORES EM COMPUTAÇÃO



ALUNO (A): _____
PROFESSOR(A): _____
DATA: / /
VALOR: 30 pontos **Nota:** _____

1. Dada uma lista de números inteiros [29, 10, 14, 37, 13], escreva um código em Python que utilize o método de ordenação Selection Sort para ordenar essa lista em ordem crescente. Em seguida, explique brevemente como o algoritmo está funcionando passo a passo.
2. Implemente o algoritmo de Selection Sort em Python para ordenar uma lista de nomes de alunos em ordem alfabética
Dada a lista: alunos = ["Carlos", "Ana", "Beatriz", "Eduardo", "Diana"]
Escreva um código que utilize o método **Selection Sort** para ordenar essa lista de forma crescente.
3. Suponha que você tenha os seguintes algoritmos para resolver o problema de encontrar o menor valor em uma lista de números inteiros. Cada algoritmo possui uma estrutura diferente de iteração, e sua tarefa é determinar a complexidade de tempo de cada um, em termos de Big O.

Algoritmo A:

```
1 def algoritmo_a(lista):
2     menor = lista[0]
3     for num in lista:
4         if num < menor:
5             menor = num
6     return menor
```

Algoritmo B:

```
def algoritmo_b(lista):
    menor = lista[0]
    for i in range(len(lista)):
        for j in range(i + 1, len(lista)):
            if lista[j] < menor:
                menor = lista[j]
    return menor
```

Algoritmo C:

```
def algoritmo_c(lista):
    return min(lista)
```

Qual é a complexidade de tempo de cada um dos algoritmos acima em termos de Big O? Selecione a resposta correta para cada algoritmo?

4. Considere o particionamento de Lomuto, uma técnica comum usada no algoritmo QuickSort para reorganizar os elementos de uma lista em torno de um **pivô**. Esse método seleciona o último elemento como pivô e posiciona todos os elementos menores que o pivô à sua esquerda e todos os elementos maiores à sua direita.

Dada a lista:

lista = [8, 3, 1, 7, 0, 10, 2]

Utilize o particionamento de Lomuto para reorganizar a lista em torno do último elemento (2, neste caso) e responda às perguntas a seguir.

Código do particionamento de Lomuto

Complete o código a seguir para que ele implemente corretamente o particionamento de Lomuto. Esse código deve colocar todos os elementos menores que o pivô à esquerda e os elementos maiores ou iguais à direita, retornando a posição final do pivô.

[particionamento_lomuto.py](#)

5. Após a execução completa do particionamento, qual será a posição final do pivô (2) na lista?
a) 0 b) 2 c) 4 d) 6

OBS: Salve a resolução da prova em um arquivo zip, mude a extensão de .py para txt onde houver código