📍 [Checkout de Presença] Módulo 5 – Noções de eficiência e complexidade de algoritmos

Nome completo: **Diego Serafim de Sousa**

Nome do/a Professor/a Tutor/a: **Samuel Benjoino Ferraz Aquino**

Título da atividade: **ALGORITMOS E PROGRAMAÇÃO II-T01-2023-2**

Vencimento: segunda, 27 nov 2023, 23:59

Nesta atividade você demonstrará, na prática, os impactos de desempenho causados pela complexidade de um algoritmo.
Para isso, você deve implementar em Python os dois algoritmos a seguir (representados em pseudocódigo):

```
Algoritmo 1: complexidade O(n²)
soma = 0
recebe um valor inteiro n
laço de 1 até n
    laço de 1 até n
        aleatorio = valor aleatório entre 1 e 1000
        soma = soma + aleatorio


Algoritmo 2: complexidade O(n³)
soma = 0
recebe um valor inteiro n
laço de 1 até n
    laço de 1 até n
        laço de 1 até n
        aleatorio = valor aleatório entre 1 e 1000
        soma = soma + aleatorio
```

Uma vez implementados esses algoritmos, execute-os utilizando os seguintes valores de n:

n = 10
n = 100
n = 1000 (não interrompa a execução)

Para cada valor de n utilizado, meça o tempo gasto de cada algoritmo utilizando a função time do Python.

⇨ O link a seguir apresenta um exemplo de como utilizar a função time: >>>

Em um editor de texto, construa uma tabela que mostre o tempo de execução de cada algoritmo para cada valor de n.

No mesmo arquivo, indique quais conclusões você obteve em relação ao impacto da complexidade de um algoritmo no seu desempenho final.
Exporte o arquivo para PDF e envie no espaço da tarefa.

Algoritmo 1

```
import random # Gera números aleatórios
import time # Usado para medir o tempo gasto

def algoritmo_1(n):
    soma = 0
    for i in range(1, n + 1): # Loop de 1 até n
        for j in range(1, n + 1): # Loop de interno 1 até n
            aleatorio = random.randint(1, 1000) # Gera número inteiro aleatório entre
            soma = soma + aleatorio # Soma o nº aleatório + a variável
    return soma # Retorna a soma total

valores_de_n = [10, 100, 1000]

for n in valores_de_n:
    inicia_time = time.time() # Início da contagem de tempo
```

```
    resultado = algoritmo_1(n) # Chama o algoritmo
    fim_time = time.time() # Fim da contagem
    tempo_algoritmo = fim_time - inicia_time # Calcula o tempo de execução

    print(f"n = {n}")
    print(f"Tempo de execução do Algoritmo 1: complexidade O(n²): {tempo_algoritmo} segundos")


"""
Resultado da Saída
└─$ python3 0-modulo5_checkout-de-presenca1.py
    n = 10
    Tempo de execução do Algoritmo 1: complexidade O(n²): 8.893013000488281e-05 segundos
    n = 100
    Tempo de execução do Algoritmo 1: complexidade O(n²): 0.006061077117919922 segundos
    n = 1000
    Tempo de execução do Algoritmo 1: complexidade O(n²): 0.5249819755554199 segundos
"""
```

Algoritmo 2

```
import random # Gera números aleatórios
import time # Usado para medir o tempo gasto

def algoritmo_2(n):
    soma = 0
    for i in range(1, n + 1): # Loop de 1 até n
        for j in range(1, n + 1): # Loop de interno 1 até n
            for k in range(1, n + 1): # Loop de interno 1 até n
                aleatorio = random.randint(1, 1000) # Gera número inteiro aleatório entre
                soma = soma + aleatorio # Soma o nº aleatório + a variável
    return soma # Retorna a soma total

valores_de_n = [10, 100, 1000]

for n in valores_de_n:
    inicia_time = time.time() # Início da contagem de tempo
    resultado = algoritmo_2(n) # Chama o algoritmo
    fim_time = time.time() # Fim da contagem
    tempo_algoritmo = fim_time - inicia_time # Calcula o tempo de execução

    print(f"n = {n}") # Exibe o valor de valores_de_n
    print(f"Tempo de execução do Algoritmo 2: complexidade O(n³): {tempo_algoritmo} segundos")

"""
Resultado da Saída
└─$ python3 0-modulo5_checkout-de-presenca2.py
    n = 10
    Tempo de execução do Algoritmo 2: complexidade O(n³): 0.0006062984466552734 segundos
    n = 100
    Tempo de execução do Algoritmo 2: complexidade O(n³): 0.4692845344543457 segundos
    n = 1000
    Tempo de execução do Algoritmo 2: complexidade O(n³): 526.4349241256714 segundos
"""
```