Fazer analise assintótica dos seguintes algoritmos

A) **Vetor x vetor**

```
def vetor_x_vetor(v1, v2):
    if len(v1) != len(v2):
        raise ValueError("Os vetores devem ter o mesmo tamanho.")
    return sum(v1[i] * v2[i] for i in range(len(v1)))
```

Polinômio de instruções 2N -> Big “O” -> O(N)

B) **Vetor x matriz**

```
def vetor_x_matriz(v, M):

  n = len(M)

  if any(len(linha) != n for linha in M):
    raise ValueError("A matriz deve ser quadrada.")
  if len(v) != n:
    raise ValueError("O tamanho do vetor deve ser igual ao tamanho da
matriz.")
  resultado = [0] * n
  for j in range(n):  # percorre colunas da matriz
    soma = 0
    for i in range(n):  # percorre linhas
      soma += v[i] * M[i][j]
    resultado[j] = soma
  return resultado
```

Polinômio de instruções $2N^2$ -> Big “O” -> $O(N^2)$

C) **Matriz x matriz**

```
def matriz_x_matriz(A, B):
  if len(A[0]) != len(B):
    raise ValueError("Número de colunas de A deve ser igual ao número
de linhas de B.")

  linhas_A = len(A)
  colunas_B = len(B[0])
  colunas_A = len(A[0])

  resultado = [[0 for _ in range(colunas_B)] for _ in range(linhas_A)]
  for i in range(linhas_A):
    for j in range(colunas_B):
      for k in range(colunas_A):
        resultado[i][j] += A[i][k] * B[k][j]
```

return resultado

Polinômio de instruções $2N^3 + M2$ -> Big “O” -> $O(N^3)$