

# Trabalho de Matemática Concreta

- Resultado do trabalho de comparação de duas listas usando métodos diferentes:
  - Comparando elemento a elemento de forma randômica
  - Comparando elemento a elemento em ordem crescente de índice

In [23]: `import matplotlib.pyplot as plt`

In [24]: 

```
def comparaSequenciasAleatoriamente():  
    from random import randint  
    global sequenciaA, sequenciaB  
  
    indicesJaSorteados = []  
    comandos = 0  
    while True:  
        indiceAleatorio = randint(0,99)  
        while indiceAleatorio in indicesJaSorteados:  
            indiceAleatorio = randint(0,99)  
        indicesJaSorteados.append(indiceAleatorio)  
        comandos += 1  
        if sequenciaA[indiceAleatorio] != sequenciaB[indiceAleatorio]:  
            break  
    return comandos
```

In [25]: 

```
def comparaSequenciasEmOrdem():  
    global sequenciaA, sequenciaB  
  
    comandos = 0  
    for indice, elemento in enumerate(sequenciaA):  
        comandos += 1  
        if elemento != sequenciaB[indice]:  
            break  
    return comandos
```

In [26]: 

```
def trocaZeroPorUm():  
    from random import randint  
    global indicesSorteados  
    global sequenciaA, sequenciaB  
  
    indiceAleatorio = randint(0,99)  
    while indiceAleatorio in indicesSorteados:  
        indiceAleatorio = randint(0,99)  
    indicesSorteados.append(indiceAleatorio)  
    sequenciaA[indiceAleatorio] = 1
```

```
In [27]: def criaSequencias():
    global sequenciaA, sequenciaB

    sequenciaA = []
    sequenciaB = []
    for i in range(0, 100):
        sequenciaA.append(0)
        sequenciaB.append(0)
```

```
In [28]: comandosComparandoEmOrdem = []
comandosComparandoAleatoriamente = []
indicesSorteados = []
while len(indicesSorteados) != 100:
    criaSequencias()
    trocaZeroPorUm()
    comandosComparandoEmOrdem.append(comparaSequenciasEmOrdem())
    comandosComparandoAleatoriamente.append(comparaSequenciasAleatoriamente())
print(comandosComparandoEmOrdem)
print(comandosComparandoAleatoriamente)
```

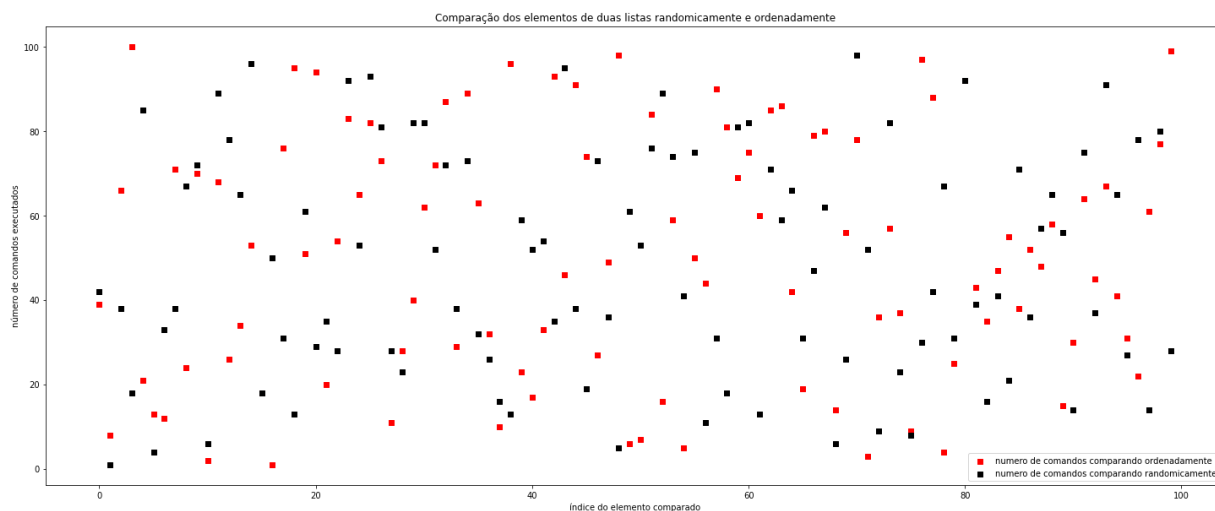
```
[39, 8, 66, 100, 21, 13, 12, 71, 24, 70, 2, 68, 26, 34, 53, 18, 1, 76, 95, 51, 94, 2
0, 54, 83, 65, 82, 73, 11, 28, 40, 62, 72, 87, 29, 89, 63, 32, 10, 96, 23, 17, 33, 9
3, 46, 91, 74, 27, 49, 98, 6, 7, 84, 16, 59, 5, 50, 44, 90, 81, 69, 75, 60, 85, 86,
42, 19, 79, 80, 14, 56, 78, 3, 36, 57, 37, 9, 97, 88, 4, 25, 92, 43, 35, 47, 55, 38,
52, 48, 58, 15, 30, 64, 45, 67, 41, 31, 22, 61, 77, 99]
[42, 1, 38, 18, 85, 4, 33, 38, 67, 72, 6, 89, 78, 65, 96, 18, 50, 31, 13, 61, 29, 3
5, 28, 92, 53, 93, 81, 28, 23, 82, 82, 52, 72, 38, 73, 32, 26, 16, 13, 59, 52, 54, 3
5, 95, 38, 19, 73, 36, 5, 61, 53, 76, 89, 74, 41, 75, 11, 31, 18, 81, 82, 13, 71, 5
9, 66, 31, 47, 62, 6, 26, 98, 52, 9, 82, 23, 8, 30, 42, 67, 31, 92, 39, 16, 41, 21,
71, 36, 57, 65, 56, 14, 75, 37, 91, 65, 27, 78, 14, 80, 28]
```

## Plotando o gráfico

```
In [29]: x = list(range(0,100))
y1 = comandosComparandoEmOrdem
y2 = comandosComparandoAleatoriamente

plt.figure(figsize=(25,10))
plt.scatter(x, y1, color='r', marker='s', label='numero de comandos comparando
sequencialmente')
plt.scatter(x, y2, color='k', marker='s', label='numero de comandos comparando
randomicamente')
plt.legend()
plt.ylabel('número de comandos executados')
plt.xlabel('índice do elemento comparado')
plt.title('Comparação dos elementos de duas listas randomicamente e
ordenadamente')
```

```
plt.show()
```



In [30]:

```
x = list(range(0,100))
y1 = comandosComparandoEmOrdem
y2 = comandosComparandoAleatoriamente

plt.figure(figsize=(25,10))
plt.plot(x, y1, color='r', marker='s', label='numero de comandos comparando
sequencialmente')
plt.plot(x, y2, color='k', marker='s', label='numero de comandos comparando
randomicamente')
plt.legend()
plt.ylabel('número de comandos executados')
plt.xlabel('índice do elemento comparado')
plt.title('Comparação dos elementos de duas listas randomicamente e
ordenadamente')

plt.show()
```

