

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 [1]: import matplotlib.pyplot as plt
```

```
In [2]: 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)
            comandos += 1
        indicesJaSorteados.append(indiceAleatorio)
        comandos += 1
        if sequenciaA[indiceAleatorio] != sequenciaB[indiceAleatorio]:
            break
    return comandos
```

```
In [3]: def comparaSequenciasEmOrdem():
    global sequenciaA, sequenciaB

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

```
In [4]: 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 [5]: def criaSequencias():
        global sequenciaA, sequenciaB

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

```
In [6]: comandosComparandoEmOrdem = []
        comandosComparandoAleatoriamente = []
        indicesSorteados = []
        while len(indicesSorteados) != 100:
            criaSequencias()
            trocaZeroPorUm()
            comandosComparandoEmOrdem.append(comparaSequenciasEmOrdem())
            comandosComparandoAleatoriamente.append(comparaSequenciasAleatoriamente())
        print(comandosComparandoEmOrdem)
        print(comandosComparandoAleatoriamente)
```

```
[93, 84, 100, 40, 24, 63, 16, 68, 85, 8, 49, 19, 25, 42, 47, 22, 67, 4, 38, 72, 55,
94, 64, 97, 58, 57, 90, 21, 1, 10, 96, 45, 98, 51, 52, 80, 7, 81, 86, 9, 69, 54, 12,
74, 23, 3, 78, 89, 30, 2, 15, 26, 34, 53, 48, 59, 99, 91, 35, 20, 83, 61, 17, 62, 8
8, 29, 79, 46, 18, 56, 32, 73, 27, 39, 11, 33, 77, 75, 31, 36, 13, 65, 66, 41, 76, 4
3, 95, 60, 44, 6, 14, 71, 82, 87, 37, 5, 50, 92, 70, 28]
[469, 16, 80, 9, 46, 209, 114, 27, 44, 139, 268, 71, 80, 61, 115, 16, 64, 32, 72, 12
6, 75, 13, 33, 80, 98, 90, 23, 166, 267, 28, 123, 282, 434, 53, 47, 180, 90, 10, 4,
189, 247, 82, 105, 6, 42, 65, 21, 75, 350, 252, 118, 765, 32, 6, 11, 109, 257, 139,
38, 97, 1, 84, 127, 204, 20, 219, 195, 47, 177, 81, 25, 192, 28, 262, 224, 22, 44,
6, 41, 197, 71, 28, 78, 33, 43, 218, 71, 63, 69, 40, 85, 181, 95, 66, 75, 5, 71, 34
0, 64, 38]
```

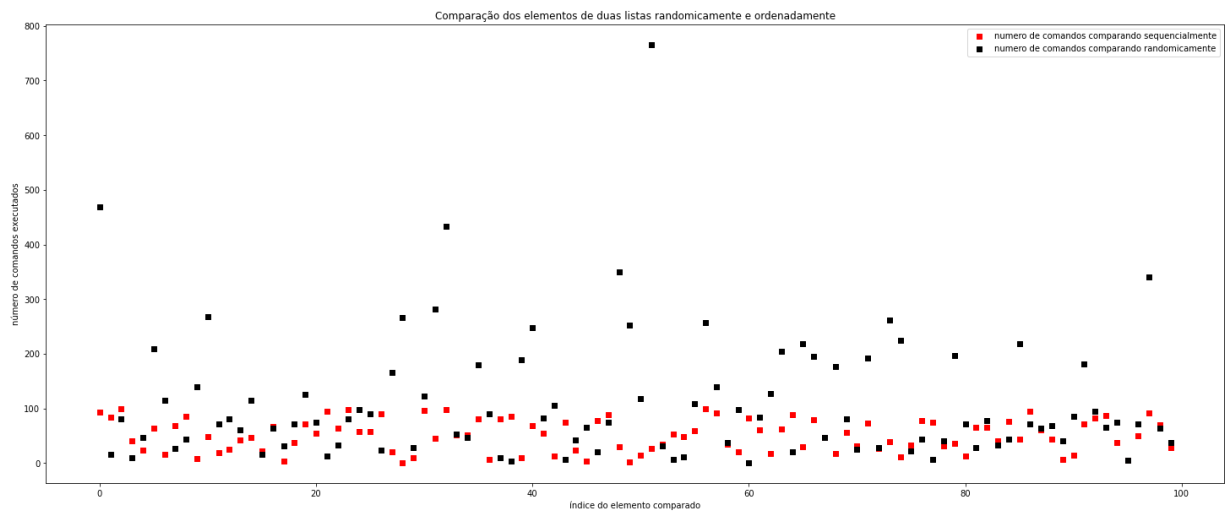
Plotando o gráfico

```
In [7]: 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 [8]:

```
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()
```

