

Missão prática - Nível 5 – Mundo 1

“Colocando tudo em ordem e guardando”

//Disciplina: Nivel5: Colocando tudo e ordem e guardando

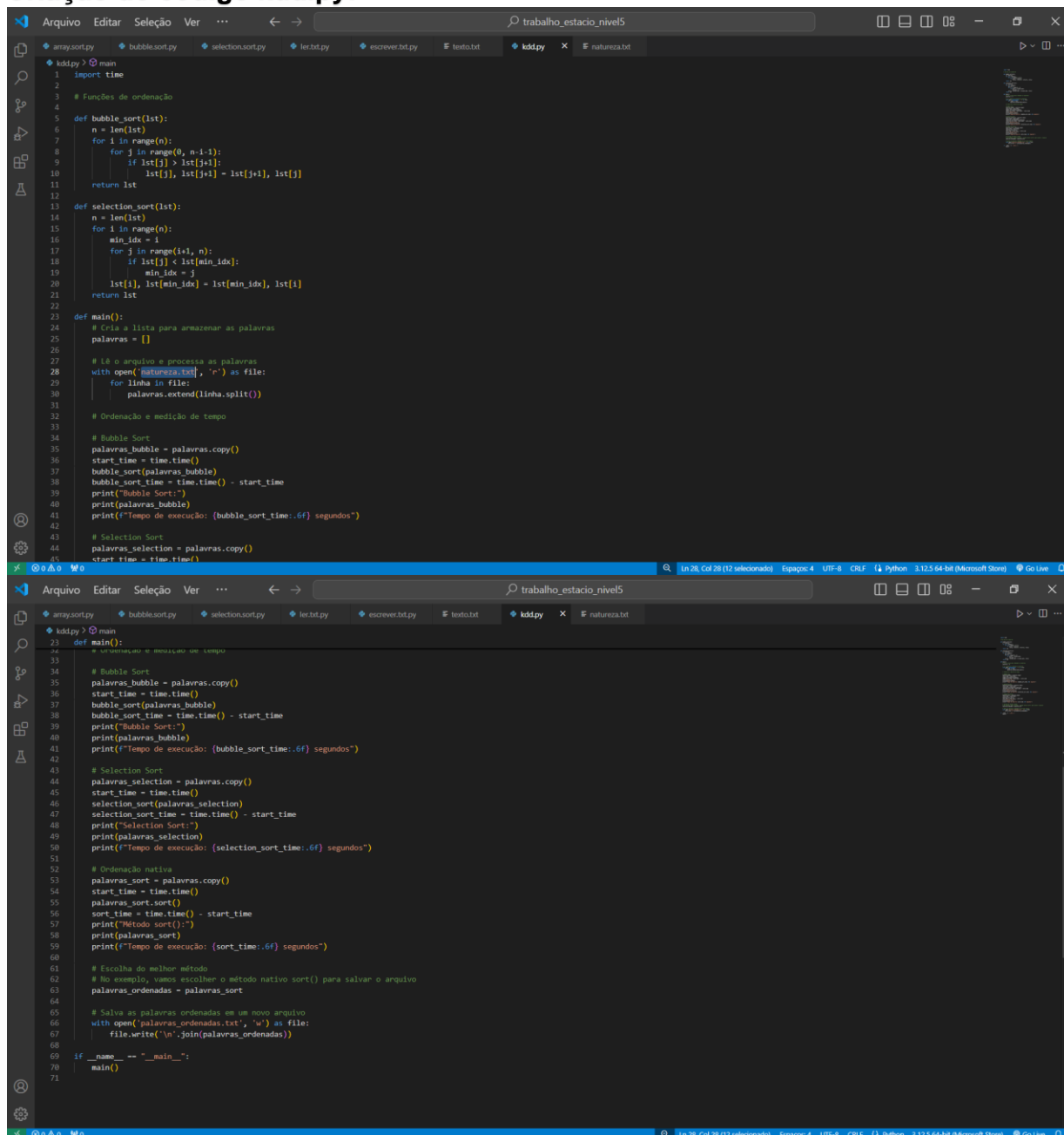
//Professor: Raul Carlos Costa Queiros

//Aluno: Rafael Lima de Medeiros

//Turma: 2024.2

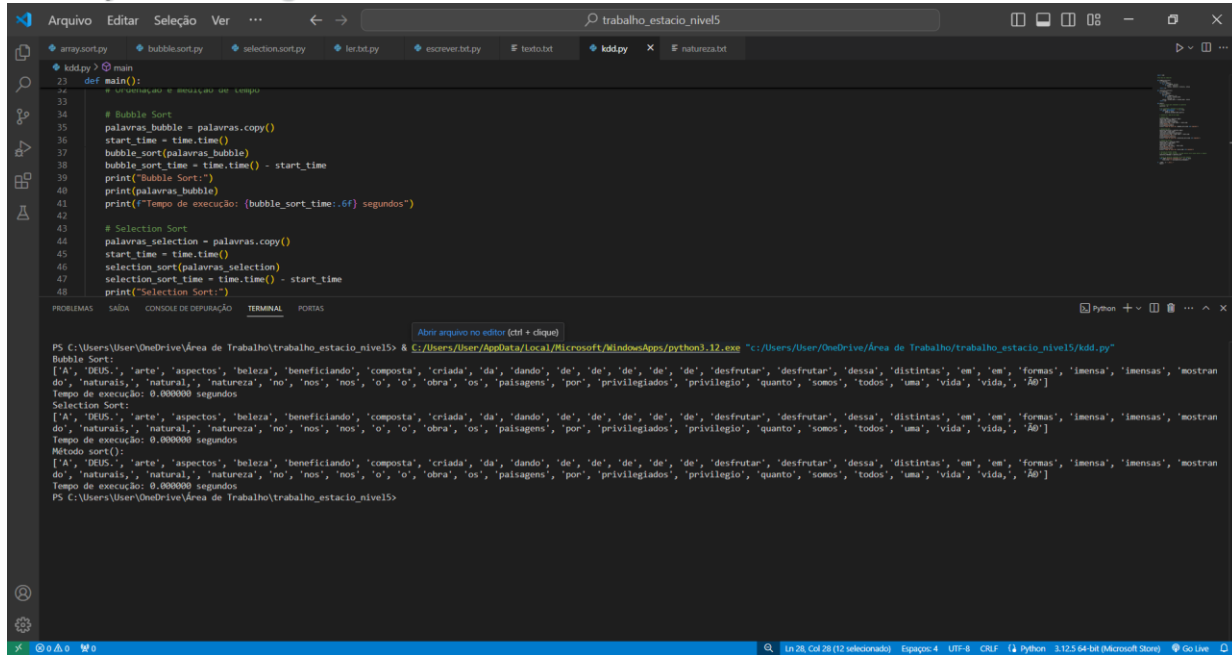
//Data atual: 25/08/2024

Criação do código kdd.py:



```
1 import time
2
3 # Funções de ordenação
4
5 def bubble_sort(lst):
6     n = len(lst)
7     for i in range(n):
8         for j in range(0, n-i-1):
9             if lst[j] > lst[j+1]:
10                 lst[j], lst[j+1] = lst[j+1], lst[j]
11     return lst
12
13 def selection_sort(lst):
14     n = len(lst)
15     for i in range(n):
16         min_idx = i
17         for j in range(i+1, n):
18             if lst[j] < lst[min_idx]:
19                 min_idx = j
20         lst[i], lst[min_idx] = lst[min_idx], lst[i]
21     return lst
22
23 def main():
24     # Cria a lista para armazenar as palavras
25     palavras = []
26
27     # Lê o arquivo e processa as palavras
28     with open('natureza.txt', 'r') as file:
29         for linha in file:
30             palavras.extend(linha.split())
31
32     # Ordenação e medição de tempo
33
34     # Bubble Sort
35     palavras_bubble = palavras.copy()
36     start_time = time.time()
37     bubble_sort(palavras_bubble)
38     bubble_sort_time = time.time() - start_time
39     print("Bubble Sort:")
40     print(palavras_bubble)
41     print(f"Tempo de execução: {bubble_sort_time:.6f} segundos")
42
43     # Selection Sort
44     palavras_selection = palavras.copy()
45     start_time = time.time()
46     selection_sort(palavras_selection)
47     selection_sort_time = time.time() - start_time
48     print("Selection Sort:")
49     print(palavras_selection)
50     print(f"Tempo de execução: {selection_sort_time:.6f} segundos")
51
52     # Ordenação nativa
53     palavras_sort = palavras.copy()
54     start_time = time.time()
55     palavras_sort.sort()
56     sort_time = time.time() - start_time
57     print("Método sort():")
58     print(palavras_sort)
59     print(f"Tempo de execução: {sort_time:.6f} segundos")
60
61     # Escolha do melhor método
62     # No exemplo, vamos escolher o método nativo sort() para salvar o arquivo
63     palavras_ordenadas = palavras_sort
64
65     # Salva as palavras ordenadas em um novo arquivo
66     with open('palavras_ordenadas.txt', 'w') as file:
67         file.write("\n".join(palavras_ordenadas))
68
69 if __name__ == "__main__":
70     main()
71
```

Execução do código:



```
Arquivo Editar Seleção Ver ... trabalho_estacio_niveis
kdd.py x natureza.txt

def main():
    # Bubble Sort
    palavras_bubble = palavras.copy()
    start_time = time.time()
    bubble_sort(palavras_bubble)
    bubble_sort_time = time.time() - start_time
    print("Bubble Sort:")
    print(palavras_bubble)
    print(f"Tempo de execução: {bubble_sort_time:.6f} segundos")

    # Selection Sort
    palavras_selection = palavras.copy()
    start_time = time.time()
    selection_sort(palavras_selection)
    selection_sort_time = time.time() - start_time
    print("Selection Sort:")

PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_niveis> & C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.12.exe "c:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_niveis\kdd.py"
Bubble Sort:
['A', 'DEUS', 'arte', 'aspectos', 'beleza', 'beneficiando', 'composta', 'criada', 'da', 'dando', 'de', 'de', 'de', 'de', 'de', 'desfrutar', 'desfrutar', 'dessa', 'distintas', 'em', 'em', 'formas', 'imensa', 'imensas', 'mostran
do', 'naturais', 'natural', 'natureza', 'no', 'nos', 'nos', 'o', 'o', 'obra', 'os', 'paisagens', 'por', 'privilegiados', 'privilegio', 'quanto', 'somos', 'todos', 'uma', 'vida', 'vida', 'Ab']
Tempo de execução: 0.000000 segundos
Selection Sort:
['A', 'DEUS', 'arte', 'aspectos', 'beleza', 'beneficiando', 'composta', 'criada', 'da', 'dando', 'de', 'de', 'de', 'de', 'de', 'desfrutar', 'desfrutar', 'dessa', 'distintas', 'em', 'em', 'formas', 'imensa', 'imensas', 'mostran
do', 'naturais', 'natural', 'natureza', 'no', 'nos', 'nos', 'o', 'o', 'obra', 'os', 'paisagens', 'por', 'privilegiados', 'privilegio', 'quanto', 'somos', 'todos', 'uma', 'vida', 'vida', 'Ab']
Tempo de execução: 0.000000 segundos
Método sort():
['A', 'DEUS', 'arte', 'aspectos', 'beleza', 'beneficiando', 'composta', 'criada', 'da', 'dando', 'de', 'de', 'de', 'de', 'de', 'desfrutar', 'desfrutar', 'dessa', 'distintas', 'em', 'em', 'formas', 'imensa', 'imensas', 'mostran
do', 'naturais', 'natural', 'natureza', 'no', 'nos', 'nos', 'o', 'o', 'obra', 'os', 'paisagens', 'por', 'privilegiados', 'privilegio', 'quanto', 'somos', 'todos', 'uma', 'vida', 'vida', 'Ab']
Tempo de execução: 0.000000 segundos
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_niveis>
```

Código kdd.py:

```
import time
```

```
# Funções de ordenação
```

```
def bubble_sort(lst):
    n = len(lst)
    for i in range(n):
        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
    return lst
```

```
def selection_sort(lst):
    n = len(lst)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if lst[j] < lst[min_idx]:
                min_idx = j
        lst[i], lst[min_idx] = lst[min_idx], lst[i]
    return lst
```

```
def main():
    # Cria a lista para armazenar as palavras
    palavras = []
```

```
# Lê o arquivo e processa as palavras
with open('natureza.txt', 'r') as file:
    for linha in file:
        palavras.extend(linha.split())
```

```
# Ordenação e medição de tempo
```

```
# Bubble Sort
palavras_bubble = palavras.copy()
```

```
start_time = time.time()
bubble_sort(palavras_bubble)
bubble_sort_time = time.time() - start_time
print("Bubble Sort:")
print(palavras_bubble)
print(f"Tempo de execução: {bubble_sort_time:.6f} segundos")
```

```
# Selection Sort
palavras_selection = palavras.copy()
start_time = time.time()
selection_sort(palavras_selection)
selection_sort_time = time.time() - start_time
print("Selection Sort:")
print(palavras_selection)
print(f"Tempo de execução: {selection_sort_time:.6f} segundos")
```

```
# Ordenação nativa
palavras_sort = palavras.copy()
start_time = time.time()
palavras_sort.sort()
sort_time = time.time() - start_time
print("Método sort():")
print(palavras_sort)
print(f"Tempo de execução: {sort_time:.6f} segundos")
```

```
# Escolha do melhor método
# No exemplo, vamos escolher o método nativo sort() para salvar o arquivo
palavras_ordenadas = palavras_sort
```

```
# Salva as palavras ordenadas em um novo arquivo
with open('palavras_ordenadas.txt', 'w') as file:
    file.write("\n".join(palavras_ordenadas))
```

```
if __name__ == "__main__":
    main()
```