Recursos de hardware:

Processador Intel(R) Celeron(R) CPU 4205U

UHD GRAPHICS 610.

2 x 4 GB de memória RAM 2400 MHz

Windows 11

Radix Sort:

Complexidade do algoritmo: A complexidade do Radix Sort usando o Counting Sort como subrotina de ordenação é O(d * (n + k)), onde d é o número de dígitos dos elementos a serem ordenados, n é o número de elementos a serem ordenados e k é o intervalo de valores possíveis para cada dígito (no caso de valores inteiros, k é igual a 10).

Isso ocorre porque o Radix Sort percorre cada dígito dos elementos a serem ordenados, realizando uma ordenação estável usando o Counting Sort em cada dígito. O Counting Sort cria um array de frequências com tamanho k (10, no caso de valores inteiros), o que adiciona um termo k na complexidade.

Tempo de execução no ambiente local: 0.102855 segundos

Quick Sort:

Complexidade do algoritmo: O Quick Sort tem complexidade O(n log n), que é uma das complexidades mais eficientes para algoritmos de ordenação. O algoritmo divide repetidamente a lista em duas metades até que cada sublista tenha apenas um elemento, o que requer log n divisões. Em seguida, o algoritmo intercala as sublistas em ordem crescente, o que requer n operações. Portanto, a complexidade do QuickSort é n log n.

Tempo de execução no ambiente local: 0.372116 segundos.

Bucket Sort:

Complexidade do algoritmo: A complexidade do Bucket Sort é O(n+k) porque ele percorre a lista de elementos a serem ordenados uma vez para colocar cada elemento em um dos baldes correspondentes, e depois percorre cada balde individualmente para ordenar seus elementos. A complexidade de percorrer a lista de elementos é O(n), enquanto a complexidade de percorrer cada balde é O(k), onde k é o número de baldes. Portanto, a complexidade total do algoritmo é O(n+k).

Tempo de execução no ambiente local: 0.37854 segundos.