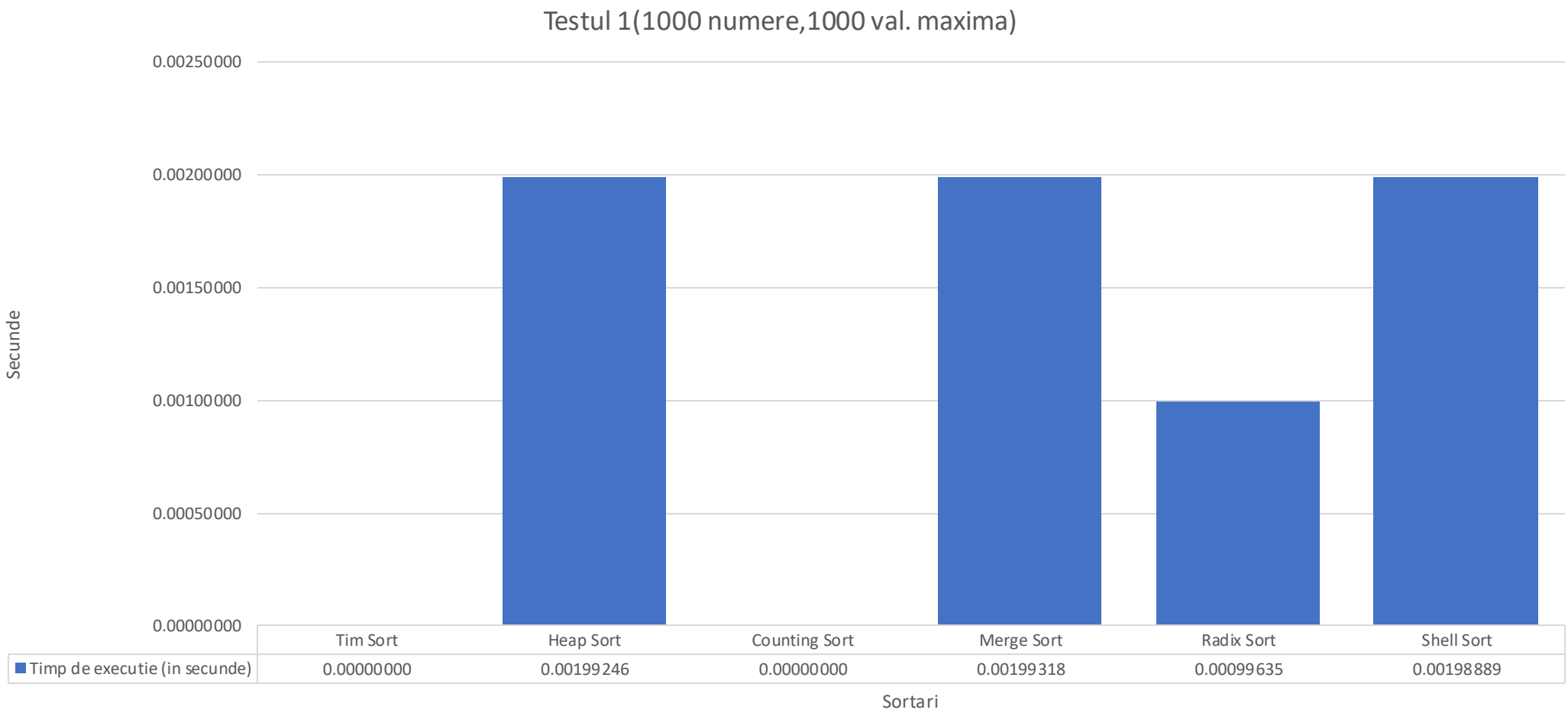
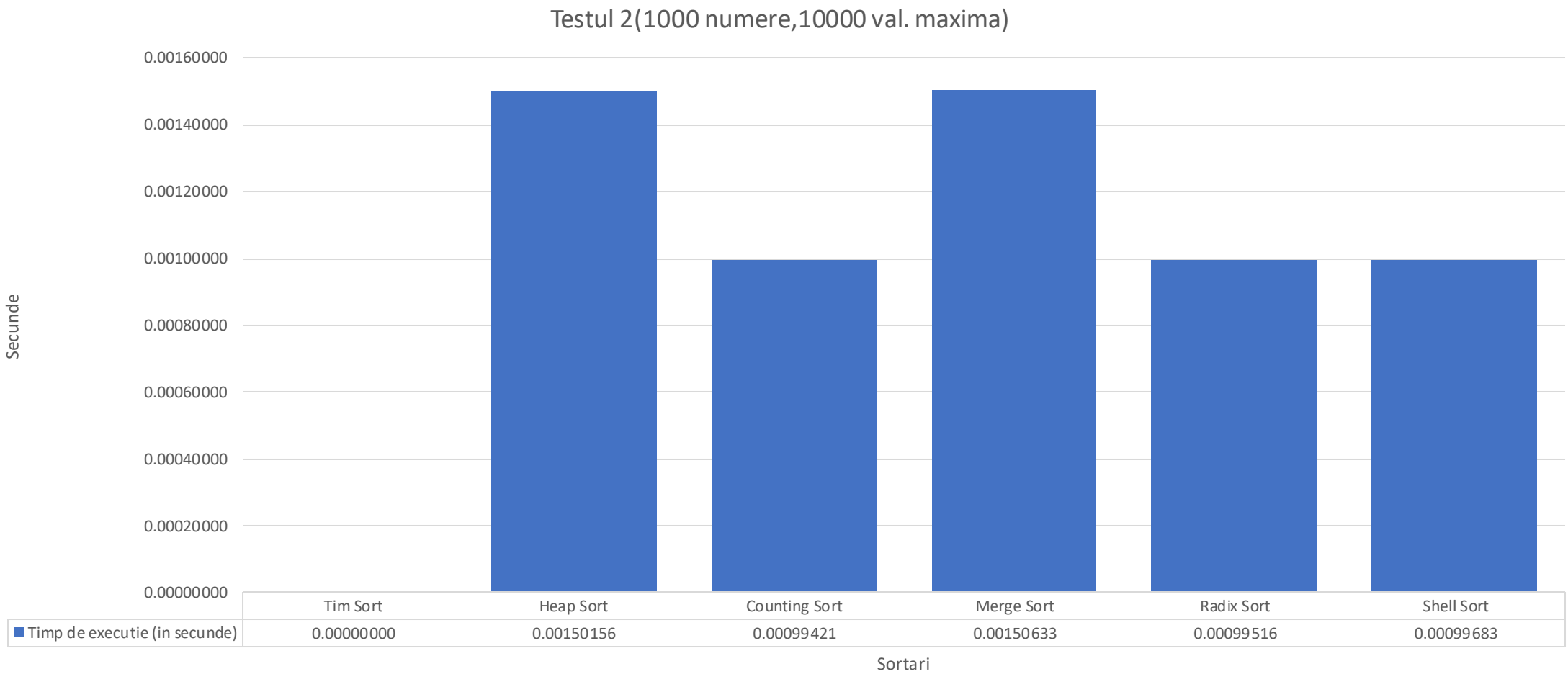
Tema 1 Structuri de date
-Sortari-

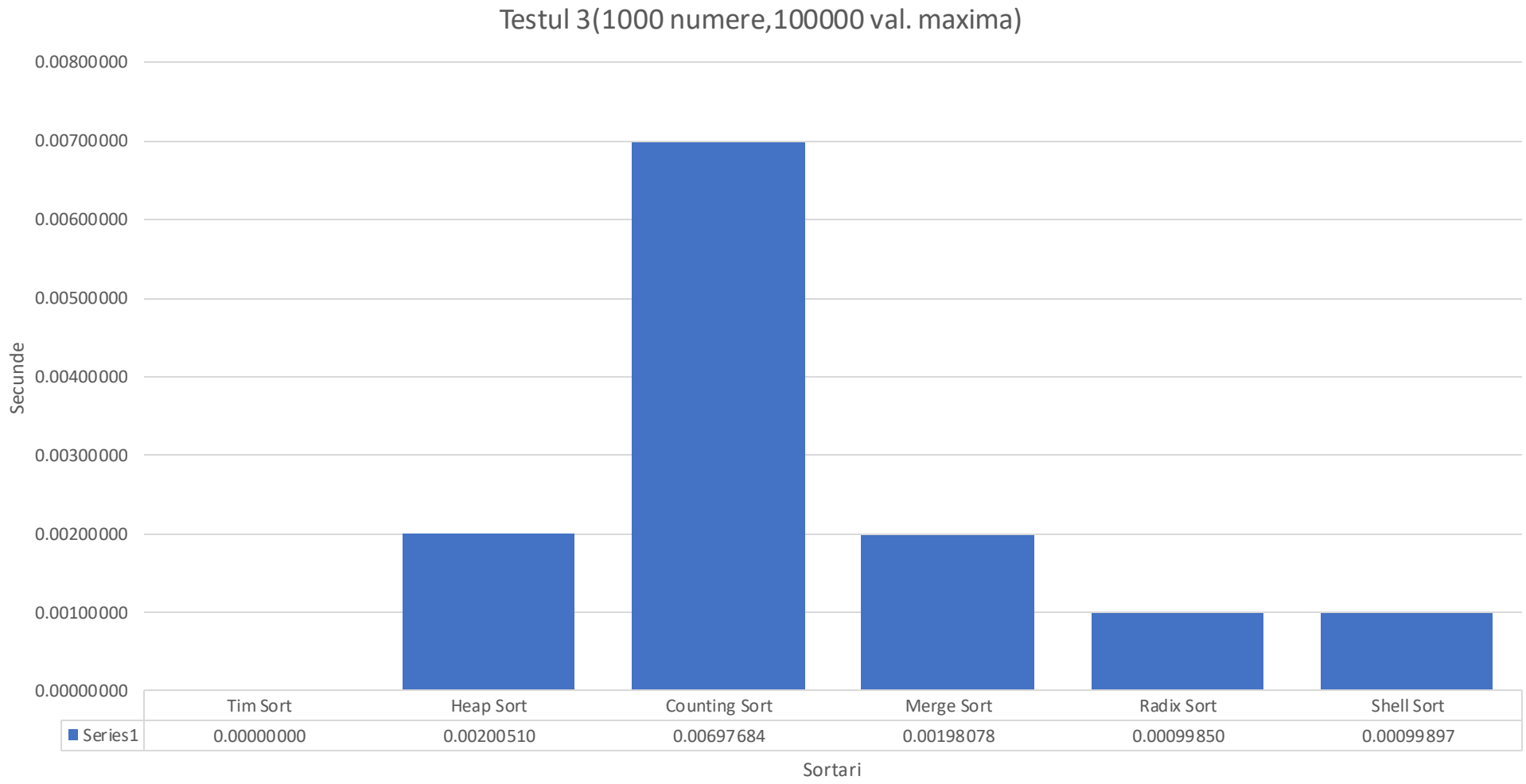
Trandafir Alexandru Ionut-grupa 133

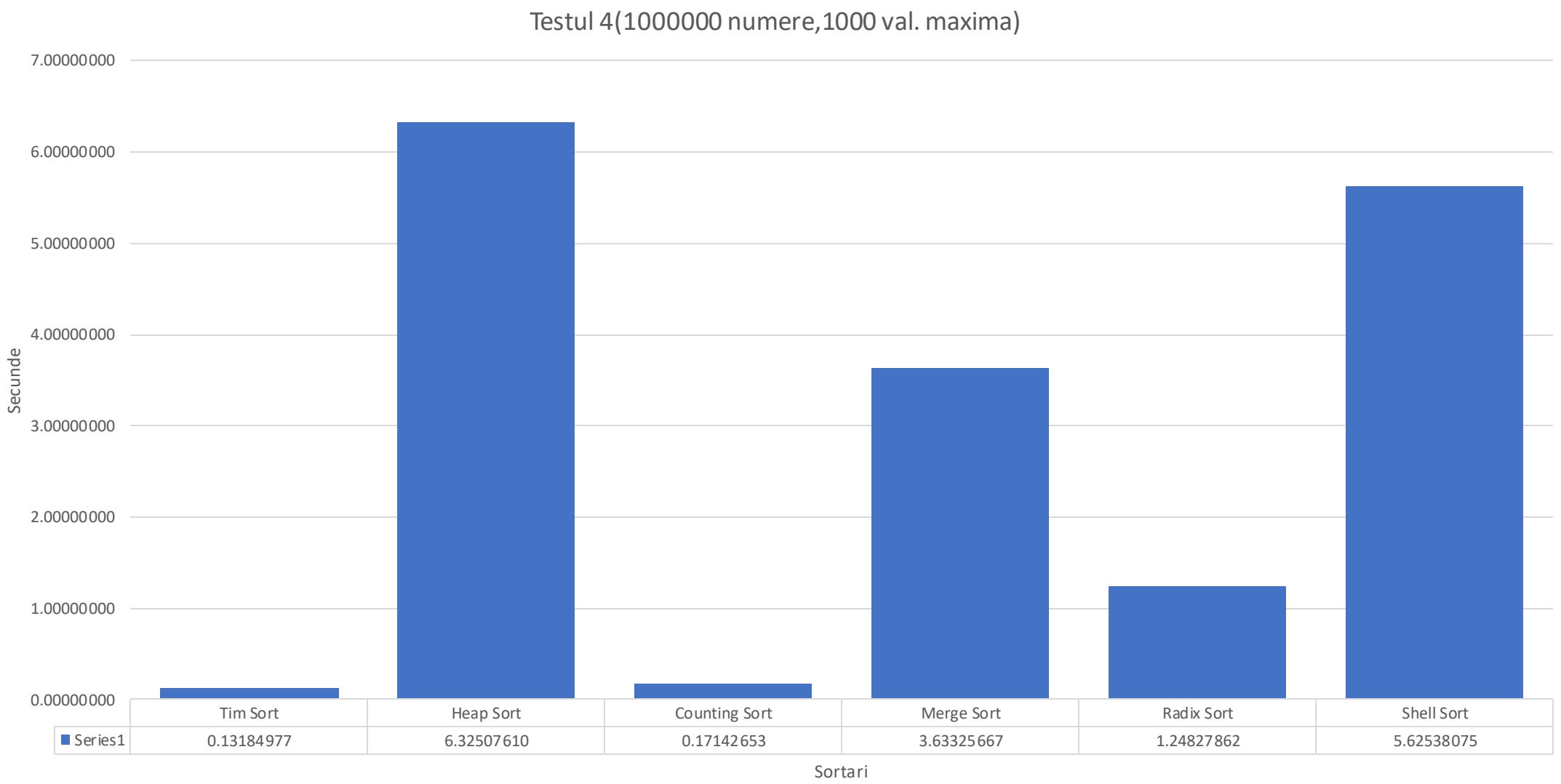
Ce teste am aplicat?

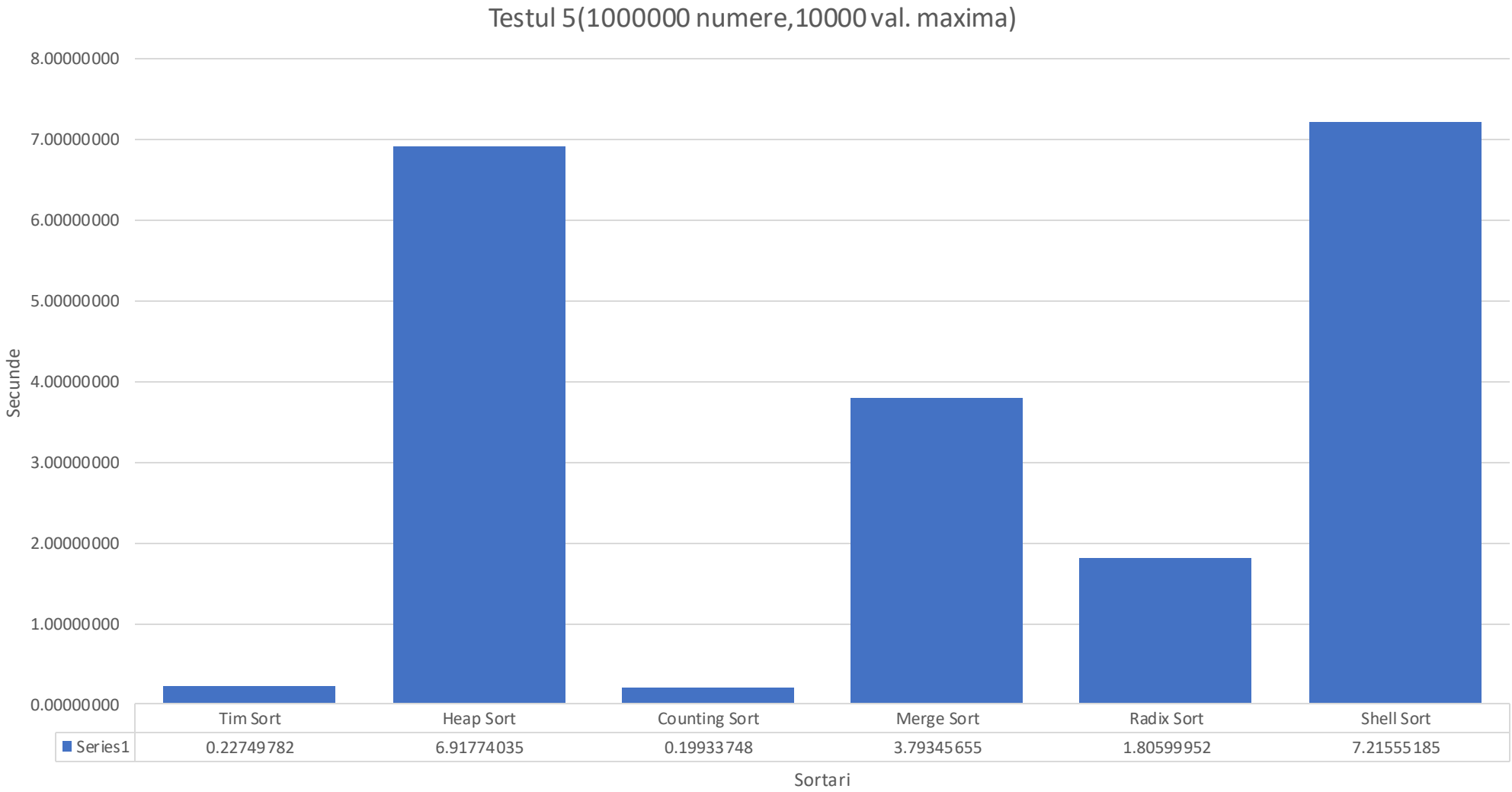
- Testul 1: 1000 numere(10^3),1000 val. Maxima
- Testul 2: 1000 numere(10^3),10000 val. Maxima
- Testul 3: 1000 numere(10^3),100000 val. Maxima
- Testul 4: 1000000 numere(10^6),1000 val. Maxima
- Testul 5:1000000 numere(10^6),10000 val. Maxima
- Testul 6: 1000000 numere(10^6),100000 val. Maxima
- Testul 7: 100000000 numere(10^8),10 val. Maxima
- Testul 8: 100000000 numere(10^8),100 val. maxima

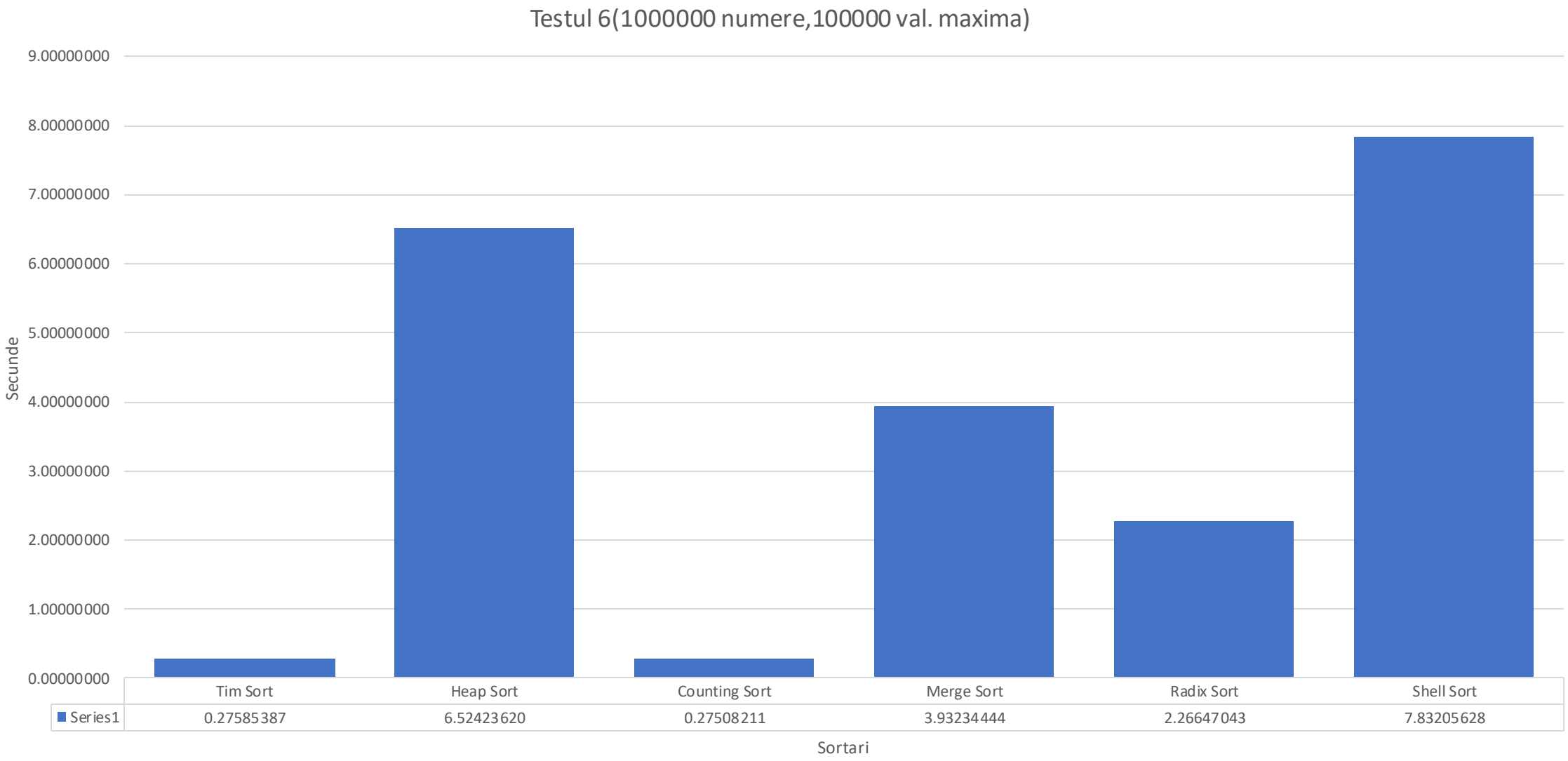

Testul 1(1000 numere,1000 val. maxima)
0.00250000
0.00200000
0.00150000
0.00100000
0.00050000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Timp de executie (in secunde)
0.00000000
0.00199246
0.00000000
0.00199318
0.00099635
0.00198889
Sortari

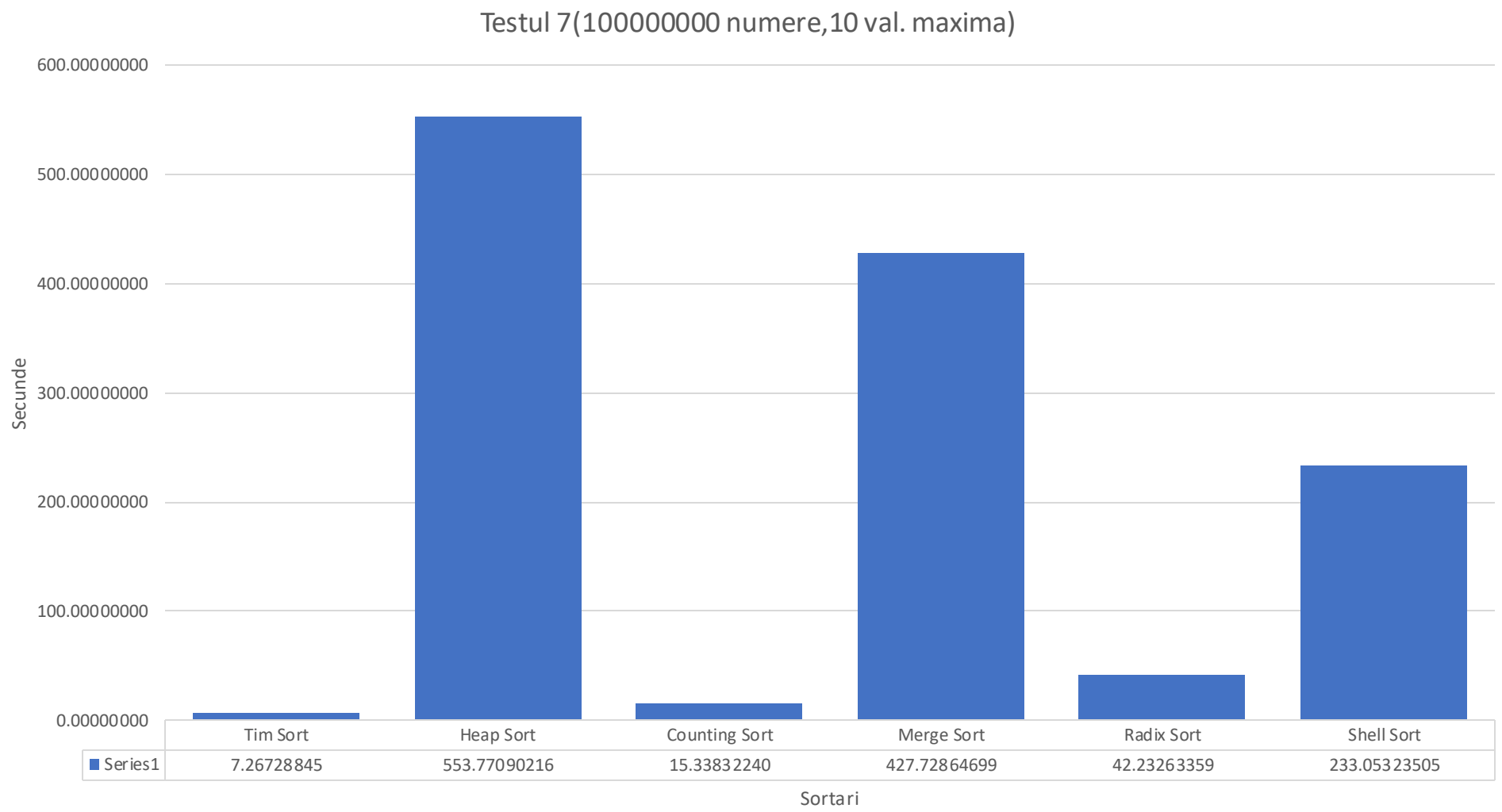

Testul 2(1000 numere,10000 val. maxima)
0.00160000
0.00140000
0.00120000
0.00100000
0.00080000
0.00060000
0.00040000
0.00020000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Timp de executie (in secunde)
0.00000000
0.00150156
0.00099421
0.00150633
0.00099516
0.00099683
Sortari

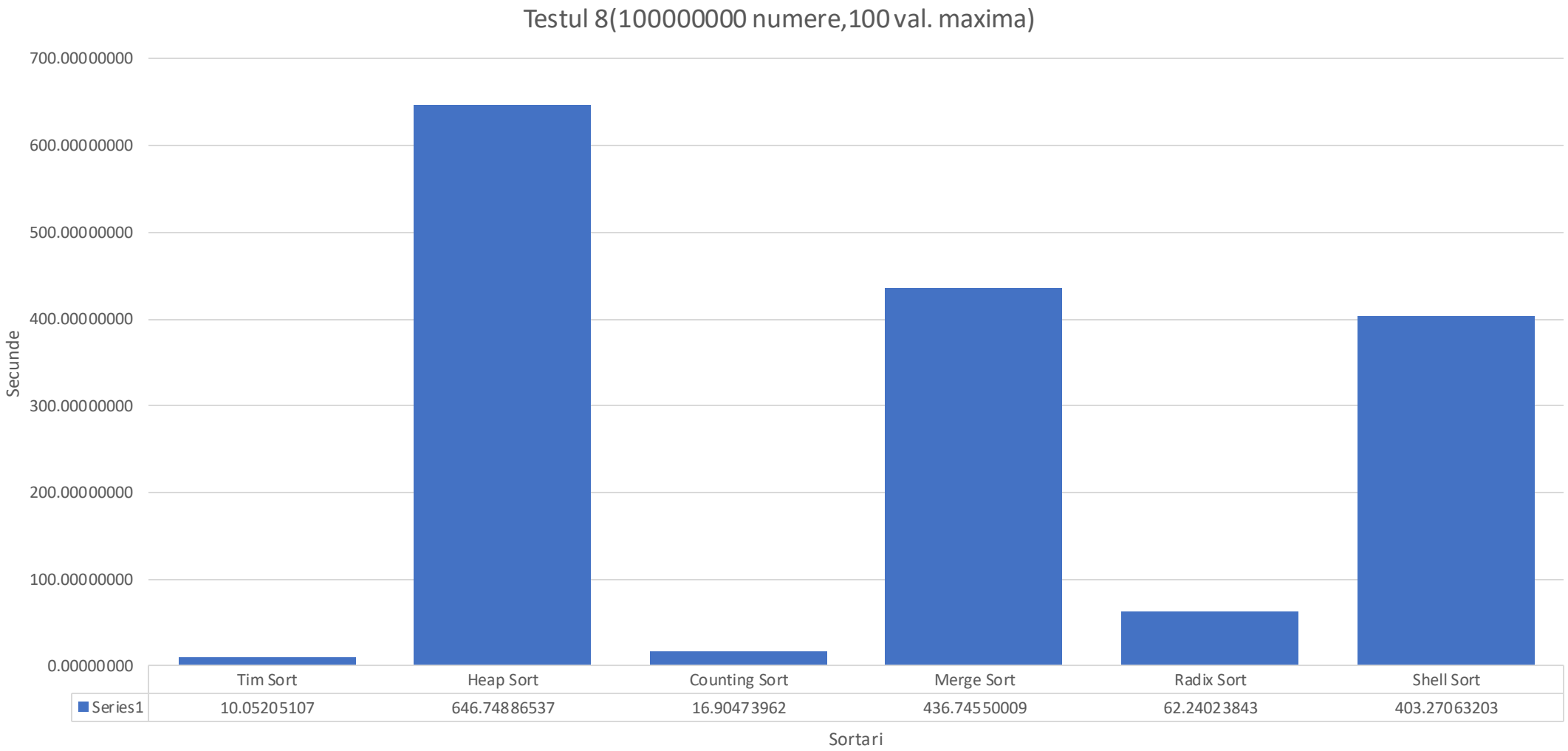

Testul 3(1000 numere,100000 val. maxima)
0.00800000
0.00700000
0.00600000
0.00500000
0.00400000
0.00300000
0.00200000
0.00100000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
0.00000000
0.00200510
0.00697684
0.00198078
0.00099850
0.00099897
Sortari


Testul 4(1000000 numere,1000 val. maxima)
7.00000000
6.00000000
5.00000000
4.00000000
3.00000000
2.00000000
1.00000000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
0.13184977
6.32507610
0.17142653
3.63325667
1.24827862
5.62538075
Sortari


Testul 5(1000000 numere,10000 val. maxima)
8.00000000
7.00000000
6.00000000
5.00000000
4.00000000
3.00000000
2.00000000
1.00000000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
0.22749782
6.91774035
0.19933748
3.79345655
1.80599952
7.21555185
Sortari


Testul 6(1000000 numere,100000 val. maxima)
9.00000000
8.00000000
7.00000000
6.00000000
5.00000000
4.00000000
3.00000000
2.00000000
1.00000000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
0.27585387
6.52423620
0.27508211
3.93234444
2.26647043
7.83205628
Sortari


Testul 7(100000000 numere,10 val. maxima)
600.00000000
500.00000000
400.00000000
300.00000000
200.00000000
100.00000000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
7.26728845
553.77090216
15.33832240
427.72864699
42.23263359
233.05323505
Sortari


Testul 8(100000000 numere,100 val. maxima)
700.00000000
600.00000000
500.00000000
400.00000000
300.00000000
200.00000000
100.00000000
0.00000000
Secunde
Tim Sort
Heap Sort
Counting Sort
Merge Sort
Radix Sort
Shell Sort
Series1
10.05205107
646.74886537
16.90473962
436.74550009
62.24023843
403.27063203
Sortari

Note

- Testele de pe grafice care au rulat mai repede decat minimul acceptat de functia time() din Python, care este de 0,00001, au primit valoarea 0.
- In urma testelor, este evident faptul ca Tim Sort, sortarea implicita din Python,este cea mai eficienta.
- Heap Sort pare a fi cel mai ineficient algoritm in urma testelor
- Counting Sort este afectat valoarea maxima
- Heap Sort si Merge Sort sunt afectate de lungimea arrayului
- Radix Sort nu pare sa aiba un pattern(este greu de generalizat), dar se descurca mai bine decat unele din celelalte sorturi, precum Heap Sort, Merge Sort si Shell Sort
- Shell Sort nu pare nici el sa aiba un pattern, dar este ineficient in comparatie cu ceilalti algoritmi