GUȚĂ GEORGE
GRUPA 252

Comparații ale timpilor de rulare între algoritmii de sortare

Merge Sort

Observăm că sortarea a n = 1000000 (10**6) este ineficientă utilizând acest algoritm, timpul fiind de 8.5349 secunde.

```
Sortarea a 100 numere cu Merge Sort s-a realizate in 0.0 secunde
Sortarea a 1000 numere cu Merge Sort s-a realizate in 0.004 secunde
Sortarea a 10000 numere cu Merge Sort s-a realizate in 0.058 secunde
Sortarea a 100000 numere cu Merge Sort s-a realizate in 0.661 secunde
Sortarea a 1000000 numere cu Merge Sort s-a realizate in 8.5349 secunde
Sortarea a 10000000 numere cu Merge Sort dureaza prea mult timp!
```

Insertion Sort

Insertion Sort este mult mai ineficient decat Merge Sort pe un large data set. Deja întâmpinăm probleme pentru n = 10000 (10**4).

```
Sortarea a 100 numere cu Insertion Sort s-a realizat in 0.012 secunde
Sortarea a 1000 numere cu Insertion Sort s-a realizat in 0.054 secunde
Sortarea a 10000 numere cu Insertion Sort s-a realizat in 5.633 secunde
Sortarea a 100000 numere cu Insertion Sort dureaza prea mult timp!
Sortarea a 1000000 numere cu Insertion Sort dureaza prea mult timp!
Sortarea a 10000000 numere cu Insertion Sort dureaza prea mult timp!
```

Shell Sort

Am abordat mai multe variante de implementare pentru Shell Sort. Fie am folosit niște secvențe, fie am calculat gap-ul prin diverse metode.

Nu am reușit să obțin un rezultat concrect folosind secvența lui Knuth, însă putem nota că Merge Sort este relativ mai rapid (cel puțin pe varianta în care am folosit Ciura’s sequence):

```
Sortarea a 100 numere cu Shell Sort s-a realizat in 0.0 secunde
Sortarea a 1000 numere cu Shell Sort s-a realizat in 0.0 secunde
Sortarea a 10000 numere cu Shell Sort s-a realizat in 0.0126 secunde
Sortarea a 100000 numere cu Shell Sort s-a realizat in 1.6863 secunde
Sortarea a 1000000 numere cu Shell Sort folosind Ciura's sequence dureaza prea mult timp!
Sortarea a 10000000 numere cu Shell Sort folosind Ciura's sequence dureaza prea mult timp!
```

GUȚĂ GEORGE
GRUPA 252

Radix Sort baza 256 vs Merge Sort – Pivot mediana din 3

```
Sortarea a 100 numere folosind Radix Sort baza 256 s-a realizat in 0.1321 secunde
Sortarea a 1000 numere folosind Radix Sort baza 256 s-a realizat in 0.0021 secunde
Sortarea a 10000 numere folosind Radix Sort baza 256 s-a realizat in 0.0142 secunde
Sortarea a 100000 numere folosind Radix Sort baza 256 s-a realizat in 0.1783 secunde
Sortarea a 1000000 numere folosind Radix Sort baza 256 s-a realizat in 2.1683 secunde
Sortarea a 10000000 numere folosind Radix Sort baza 256 s-a realizat in 24.3978 secunde

Sortarea a 100 numere folosind Quick Sort - Pivot mediana din 3 s-a realizat in 0.0 secunde
Sortarea a 1000 numere folosind Quick Sort - Pivot mediana din 3 s-a realizat in 0.002 secunde
Sortarea a 10000 numere folosind Quick Sort - Pivot mediana din 3 s-a realizat in 0.0243 secunde
Sortarea a 100000 numere folosind Quick Sort - Pivot mediana din 3 s-a realizat in 0.3088 secunde
Sortarea a 1000000 numere folosind Quick Sort - Pivot mediana din 3 s-a realizat in 4.9124 secunde
Sortarea a 10000000 numere folosind Quick Sort - Pivot mediana dureaza prea mult timp!
```

Observăm performanta lui Radix Sort.