

## Napomena

Uz zadatak je priložena datoteka *main.py* koja se može koristiti za testiranje zadatka. Algoritam za sortiranje implementirati u posebnom modulu *sort.py*.

## Zadaci

1. Kreirati liste od 10, 100, 1 000, 10 000 i 100 000 elemenata. Jedan element liste predstavlja rečnik koji sadrži dva ključa: *ime* i *index*. Ime kreirati kao skup nasumično izabranih karaktera dužine 5, a indeks kao nasumično odabranu celobrojnu vrednost između 0 i 99 (duplikati su dozvoljeni). Primer liste:

```
[{'ime': 'fcuzd', 'index': 79}, {'ime': 'orsfg', 'index': 18}, {'ime': 'voojm', 'index': 78}]
```

2. Implementirati **Bucket-sort** algoritam tako da elemente liste sortira po ključu *index*. Napomena: kao rezultat je potrebno ispisati listu sa sortiranim rečnicima.
3. Izmeriti vreme izvršavanja algoritma nad listama kreiranim u prvom zadatku.
  - a. Komentarisati zavisnost vremena izvršavanja u odnosu na dužine liste.
  - b. Kolika je vremenska složenost algoritma *Bucket-sort*? Od čega zavisi ova složenost? Kada se dobija najbolje, a kada najviše vreme izvršavanja?
  - c. Da li *Bucket-sort* ima bolje ili gore vreme izvršavanja u odnosu na *Insertion-sort*? Da li ima bolje ili gore vreme izvršavanja u odnosu na *Merge-sort*?

(Odgovore napisati u komentaru u kodu.)