

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 torku koji sadrži dve vrednosti. Prvi element torke kreirati kao skup nasumično izabranih karaktera dužine 5, a drugi kao nasumično odabranu celobrojnu vrednost (duplikati su dozvoljeni). Primer liste:

```
[('wkczej', 44), ('diiqr', 14), ('ddpjp', 52), ('ozqdy', 37), ('gqexq', 51), ('sozwm', 25)]
```

2. Implementirati **Counting-sort** algoritam tako da elemente liste sortira po drugom elementu torke (celobrojna vrednost) i to u opadajućem redosledu!
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 *Counting-sort*? Koja je mana algoritma?
 - c. Da li *Counting-sort* ima bolje ili gore vreme izvršavanja u odnosu na *Quicksort*? Da li ima bolje ili gore vreme izvršavanja u odnosu na *Bubblesort*? (Odgovore napisati u komentaru u kodu.)