Zadatci za ponavljanje

RJEŠENJA

1. Vremenska složenost je proporcionalno ovisna o vremenu trajanja izvođenja algoritma i ukupnom broju operacija koje se izvode u tom algoritmu.

2. Metoda `time()` kao rezultat daje broj sekundi od početka računalne epohe, a izražena je u globalnom *UCT* vremenu, dok `clock()` daje stvarno procesorsko vrijeme u sekundama od početka određenog procesa od prvog poziva neke programske funkcije.

3.

```
>>> from time import *
>>> t = localtime()
>>> t[0]
2014
>>> t[2]
6

>>> t[7]
6
```

4. **a.** 10 **b.** 21 **c.** -2 **d.** 72 **e.** 1 **f.** -9 **g.** 6 **h.** 1

5. Primjer primjene rekurzivne funkcije je izračunavanje Fibbonacijevih brojeva. No, kod primjene rekurzije događa se da više puta izračunavamo funkciju s jednakom parametrima te će sama primjena rekurzije imati za posljedicu nešto sporiji algoritam. Problem se rješava primjenom spremanja jednom izračunanih vrijednosti u posebnu listu.

6. **a.** 7 **b.** 2

7. **a.** 9 **b.** 4

8. **a.** $O(1)$ — konstantna **b.** $O(n)$ — linearna **c.** $O(n)$ — linearna
d. $O(n)$ — linearna **e.** $O(n^2)$ — kvadratna **f.** $O(n^2)$ — kvadratna