

Tartalomjegyzék

- 1 Szótárak
kezelése
- 2 Osztók száma
- 3 A $3n+1$
probléma
- 4 Rekurzív
algoritmusok
 - ◆ 4.1
Fibonacci

Szótárak kezelése

- Legyen egy *gyumolcs_arak* nevű szótárunk, a következő kulcs-érték párokkal:

'alma': 150
'szilva': 190
'ananász': 450
'banán': 300

- És legyen egy másik, *vasarlas* nevű szótár, amely azt tárolja, miből mennyit vettünk:

'banán': 0.6
'alma': 1.5
'ananász': 2

- Írj Sage függvényt (legyen a neve *ar_szamol*), amely megkapja a fenti két szótárat (első paramétere legyen az árakat tartalmazó), és kiszámolja, hogy mennyit kell fizetnünk a gyümölcsökért!

Osztók száma

Írd át az ismétlésben felírt **titok** függvényt (amiről ugye már tudjuk, hogy kiírja az adott szám osztóit) úgy, hogy ne írjon ki semmit, csak adja vissza, hogy hány osztója van a kapott számnak.

A $3n+1$ probléma

A híres $3x+1$ probléma (Collatz-sejtés): végy egy számot, ha páratlan, szorozd meg 3-mal és adj hozzá 1-et, ha páros, oszd el 2-vel. Az állítás, hogy így bármilyen pozitív egész számból indulva elbb-utóbb eljutunk 1-ig.

Írj Sage függvényt, amely x -et kap bemenetként, és sorban kiírja a lépéseket 1-ig!

Rekurzív algoritmusok

```
def sorozat(n, a, d):
    if n == 0:
        return a
    else:
        return sorozat(n - 1, a, d) * d
```

Fibonacci

Írjatok rekurzív függvényt ami kiszámolja az **n**-edik Fibonacci számot.