

## Tartalomjegyzék

- 1 Ismétlés, és gyakorlás: Egyszer? adatszerkezetek, algoritmusok, függvények
  - ◆ 1.1 Mese
  - ◆ 1.2 Szótárak kezelése
- 2 Összetettebb feladatok
  - ◆ 2.1 Tökéletes számok
  - ◆ 2.2 Szószámláló, szótár segítségével
  - ◆ 2.3 Gyök számítás intervallum felezéssel
  - ◆ 2.4 Buborék rendezés
  - ◆ 2.5 Szigetek megszámlálása

## Ismétlés, és gyakorlás: Egyszer? adatszerkezetek, algoritmusok, függvények

### Mese

- Ha valamit nem tudunk kiszámolni, megkérdezzük a gépt?!
- Algoritmusok, programozási nyelvek, implementáció
- Egyszer? adattípusok, és összetett adatszerkezetek
- Függvények
- Diót szedünk, fűvet nyírunk, porszívózunk, Legózunk mert jó!
- [https://docs.google.com/document/d/1sn7MQl8hyO0ixGrzFWusmmqFbv1x\\_rwyDEP2PFUHpXA/edit](https://docs.google.com/document/d/1sn7MQl8hyO0ixGrzFWusmmqFbv1x_rwyDEP2PFUHpXA/edit)

### 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, mib?l mennyit vettünk:

'banán': 0.6  
'alma': 1.5  
'ananász': 2

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

## Összetettebb feladatok

### Tökéletes számok

- Egy szám tökéletes szám, ha megegyezik a nála kisebb osztóinak az összegével, pl. a 6 tökéletes szám.
- Írj függvényt, amely egy *a* számról eldönti, hogy az tökéletes szám-e.
- Írj függvényt, amely 1-t?l *n*-ig összegy?jti és egy listában visszaadja a talált tökéletes számokat.

## Szószámláló, szótár segítségével

- Készíts függvényt ami a bemenetként adott szöveget feldolgozza olyan módon, hogy a bemenetként adott szótárba minden,

a szövegben szereplő szót az előfordulásának számával beilleszti a szótárba.

- Segítség: `str.split([sep[, maxsplit]]) -> [str, str, ...]`

## Gyök számítás intervallum felezéssel

- Készíts függvényt ami a bemenetként kapott kifejezés gyökét próbálja megtalálni a szintén paraméterként megadott intervallumban

(az alapértelmezett intervallum legyen  $[-5,5]$ ) intervallum felezéssel.

## Buborék rendezés

- A buborék rendezés: <http://hu.wikipedia.org/wiki/Bubor%C3%A9krendez%C3%A9s>
- Készíts függvényt ami megvalósítja a rendezést.
  - ◆ Egy lista elemeit rendezzük a buborék rendezési algoritmus segítségével.
  - ◆ Opcionálisan adjuk meg bemenetként, hogy növekvő vagy csökkenő legyen a rendezés sorrendje.

## Szigetek megszámlálása

- Készíts függvényt, ami a bemenetként kapott mátrixban, a szomszédos 1-eseket szigetnek véve visszaadja a térképen található szigetek számát.
- A matrix eleme 1 vagy 0 lehet.
- A mátrix mindig azonos oldalhosszúságú, de bármekkora lehet.
- A szomszédos elemek csak a vízszintesen vagy függőlegesen szomszédos elemeket értjük, átlósan nem értelmezett.
- Példa bemenet: `[[0,1,1,0],[0,1,0,0],[0,0,0,1],[0,0,1,1]]`