

Tartalomjegyzék

- 1 Feladatok
 - ◆ 1.1
Függvényekről
általában
 - ◇ 1.1.1
1.
lista
összeg
 - ◇ 1.1.2
2.
függvény
meghívás
 - ◇ 1.1.3
3.
Átlag
 - ◇ 1.1.4
4. jó
zárójelezés
 - ◇ 1.1.5
5.
részlisták
 - ◆ 1.2 Gráfok
 - ◇ 1.2.1
1.
Szomszédsági
mátrix
 - ◇ 1.2.2
2.
Legnagyobb
fokszám

Feladatok

Függvényekről általában

1. lista összeg

Hívjunk fának egy objektumot ha szám, vagy ha fák listája. Tehát például [0], [1] és [2] fák, mert számok, és így [0, 1, 2] is fa, mert fák listája. Ugyanezért [0, 1, [0, 1, 2], 2] is fa, és [0, [0, 1, 2], [0, [0, 1, 2], 1, [0, 1, 2], 2], 2] is az.

Írjon egy sumtree() nevű függvényt, amely egy fában szereplő számok összegét adja vissza. Például a fenti utolsó fára 14-et.

2. függvény meghívás

Definiáljon egy kétargumentumú apply() függvényt, ami az első argumentumát (egy egyargumentumú függvény) alkalmazza a második argumentumára, és az eredményt adja vissza.

3. Átlag

Írjunk egy `atlag()` nevű függvényt, melynek tetszőlegesen sok bemenete van és kimenetnének a bemeneti számok átlagát adja vissza. Kezeljük le azt az esetet ha a bemenetek között van nem szám típusú objektum.

4. jó zárójelezés

Írjunk egy függvényt, aminek a bemenete egy string és eldönti, hogy jól van-e zárójelezve, vagyis, hogy minden nyitó '(' zárójel után következik-e megfelelően egy ')' zárójel valahol a stringben.

5. részlisták

Írjon egy `sublists()` nevű függvényt, amely az argumentumaként adott

lista összes részlistájának listáját adja vissza (tetszőleges sorrendben). (Itt most l1 részlistája l2-nek, ha kalkulus értelemben részsorozata, azaz ha l1 minden tagja szerepel l2-ben, mégpedig ugyanabban a sorrendben.) Például:

```
sublists([])
[[]]
sublists([1])
[[1], []]
sublists([1,2])
[[1, 2], [1], [2], []]
sublists([1,2,3])
[[1, 2, 3], [1, 2], [1, 3], [1], [2, 3], [2], [3], []]
```

Gráfok

1. Szomszédsági mátrix

Írjuk a `matrix()` nevű függvényt, aminek a bemenete egy természetes szám, ez a gráfunk csúcsainak a száma, a másik egy lista, amiben tuple-k vannak, egy tuple azt jelenti, hogy két csúcs között fut él. A `matrix` függvény térjen vissza a gráf szomszédsági mátrixával, azaz, ha az *i* és *j* csúcs össze van kötve, akkor a mátrix *i*-edik sora *j*-edik oszlopa legyen 1, továbbá *j*-edik sora és *i*-edik oszlopa is, egyébként pedig 0 legyen. Futtassátok le a `matrix(8, [(1,7),(8,7),(6,7),(2,3),(5,4),(2,5),(1,6),(4,7)])` példára.

2. Legnagyobb foksám

Írjunk egy `degree()` nevű függvényt, aminek ugyanazok a paraméterei, mint a `connect()` függvénynek. A függvény térjen vissza a legnagyobb foksám értékével.