

# resitev

January 28, 2024

Marsovske ladje, vemo, izgledajo kot krogi, ki vsebujejo kroge, ki vsebujejo kroge... V tej domači nalogi boste dobili podatke o ladjah v obliki slovarja. Ključni slovarja so krogi, ki vsebujejo vsaj en notranji krog, pripadajoče vrednosti pa so sezname neposredno vsebovanih notranjih krogov (ne pa tudi krogov znotraj teh krogov). Če krog ne vsebuje nobenega kroga, ne nastopa kot ključ v slovarju.

Vsi krogi so predstavljeni s trojkami ( $x$ ,  $y$ ,  $r$ ).

## 0.1 Obvezna naloga

Kroge, ki ne vsebujejo nobenih drugih krogov, so naši protimarsovski obveščevalci poimenovali “mozolji”. Napiši funkcijo `stevilo_mozoljev(krog, hierarhija)`, ki prejme nek krog in hierarhijo krogov. Vrniti mora število mozoljev znotraj tega kroga.

Če krog ne vsebuje nobenega kroga, potem je cel krog v bistvu en sam mozolj. V tem primeru mora funkcija zato vrniti 1.

Poleg tega napiši funkcijo `povrsina_mozoljev(krogi, hierarhija)`, ki vrne skupno površino “mozoljev” znotraj podanega kroga. Če krog ne vsebuje nobenega kroga, funkcija vrne površino tega kroga.

### 0.1.1 Rešitev

Če kroga ni v hierarhiji, vrnemo 1, ker gre za mozolj. Sicer vse “otroke” vprašamo, koliko mozoljev imajo in to seštejemo.

```
[1]: def stevilo_mozoljev(krog, hierarhija):  
    if krog not in hierarhija:  
        return 1  
    stevilo = 0  
    for podkrog in hierarhija[krog]:  
        stevilo += stevilo_mozoljev(podkrog, hierarhija)  
    return stevilo
```

Ali, krajše:

```
[2]: def stevilo_mozoljev(krog, hierarhija):  
    if krog not in hierarhija:  
        return 1  
    return sum(stevilo_mozoljev(podkrog, hierarhija) for podkrog in  
↳ hierarhija[krog])
```

Druga funkcija je na las podobna, le namesto enic seštevamo površine.

```
[3]: def površina_mozoljev(krog, hierarhija):  
    if krog not in hierarhija:  
        x, y, r = krog  
        return math.pi * (r ** 2)  
  
    vsota = 0  
    for i in hierarhija[krog]:  
        vsota = vsota + površina_mozoljev(i, hierarhija)  
    return vsota
```

Oziroma:

```
[4]: def površina_mozoljev(krog, hierarhija):  
    if krog not in hierarhija:  
        return pi * krog[2] ** 2  
    return sum(površina_mozoljev(podkrog, hierarhija) for podkrog in  
↳ hierarhija[krog])
```

Pri funkcijah teh tipov so študenti pogosto preveč previdni in naredijo tole:

```
[5]: def površina_mozoljev(krog, hierarhija):  
    if krog not in hierarhija:  
        x, y, r = krog  
        return math.pi * (r ** 2)  
  
    vsota = 0  
    for i in hierarhija[krog]:  
        if i not in hierarhija:  
            x, y, r = i  
            vsota += math.pi * (r ** 2)  
        else:  
            vsota = vsota + površina_mozoljev(i, hierarhija)  
    return vsota
```

if znotraj zanke - in vse, kar je v njem - je nepotrebno. Tudi če kroga ni v hierarhiji, lahko zanj funkcija pokliče samo sebe in za krog bo poskrbel (natančno takšen) if na začetku funkcije.

Takšno programiranje ni napačno, le to kaže, da študent ni pomislil za en korak rekurzije naprej.

## 0.2 Dodatna naloga

Napiši funkcijo `stevilo_edincev(krog, hierarhija)`, ki za podani krog vrne število vseh krogov znotraj njega, ki (neposredno) vsebujejo natančno en krog. (Ta krog lahko potem sicer sam vsebuje nove kroge, morda celo več njih).

Naloga je ekvivalentna tej, da bi v neki rodbini raziskovali, koliko članov rodbine ima enega samega otroka (ta pa jih ima potem lahko spet več).

### 0.2.1 Rešitev

Takole.

```
[7]: def stevilo_edincev(krog, hierarhija):  
    if krog not in hierarhija:  
        return 0  
    if len(hierarhija[krog]) == 1:  
        stevilo = 1  
    else:  
        stevilo = 0  
    return stevilo + sum(stevilo_edincev(podkrog, hierarhija)  
                        for podkrog in hierarhija[krog])
```

Ali pa, z malo več trikov, takole:

```
[8]: def stevilo_edincev(krog, hierarhija):  
    notranji = hierarhija.get(krog, [])  
    return (len(notranji) == 1) + sum(stevilo_edincev(krog, hierarhija) for  
    ↪ krog in notranji)
```