Kaj je kaj

- Oglej si naslednji rekurzivni program:

```
def RazvozlajMe(osnova, meja) :

  #osnova in meja sta nenegativni števili

  if osnova > meja :

    return -1

  else :

    if osnova == meja :

      return 1

    else :

      return osnova * RazvozlajMe(osnova + 1, meja)
```

- Kateri del metode RazvozlajMe je ustavitveni pogoj?
- Kje se izvede rekurzivni klic?
- Kaj izpišejo sledeči stavki:
  - `RazvozlajMe(14,10)`
  - `RazvozlajMe(4,7)`
  - `RazvozlajMe(0,0)`

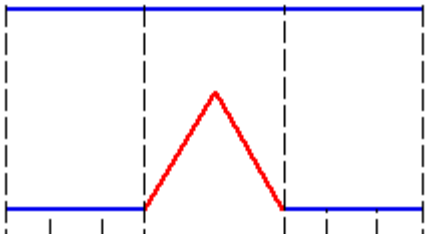
Osnove rekurzije

- Dana je rekurzivna metoda. Preberi jo in ugotovi, kaj dela.

```
def kajDelam(stevec) :
  if(stevec <= 0) :
     return ""
  else :
     return "" + str(stevec) + ", " +
          kajDelam(stevec - 1)
```

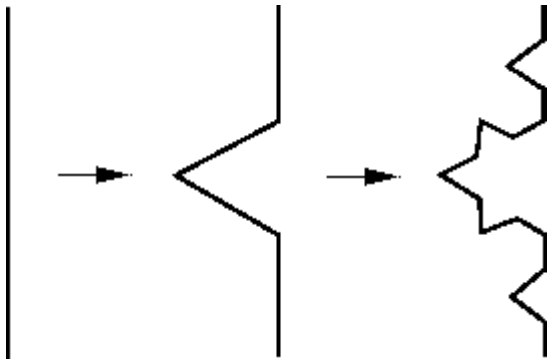
- Metodo nato prepiši tako, da bo vrnila niz s števili v obratnem vrstnem redu kot prvotna.

Črta dobi mozolje

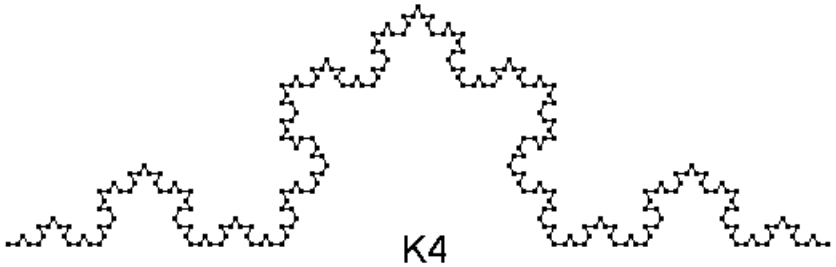
- Zvečer je bila črta še čisto normalna. Lepo gladka je potekala od točke A do točke B.
- A zjutraj se je zbudila s čudnim občutkom. Odtavala je pred ogledalo in groza! Ni bila več lepo gladka. Nad njeno srednjo tretjino se je bohotil mozolj.
- Ampak kakšen – špičast, trikoten z robovi kar take dolžine, kot je bila  lnje črte.

Črta dobi mozolje

- In naslednji dan …
- Proces se ni ustavil!

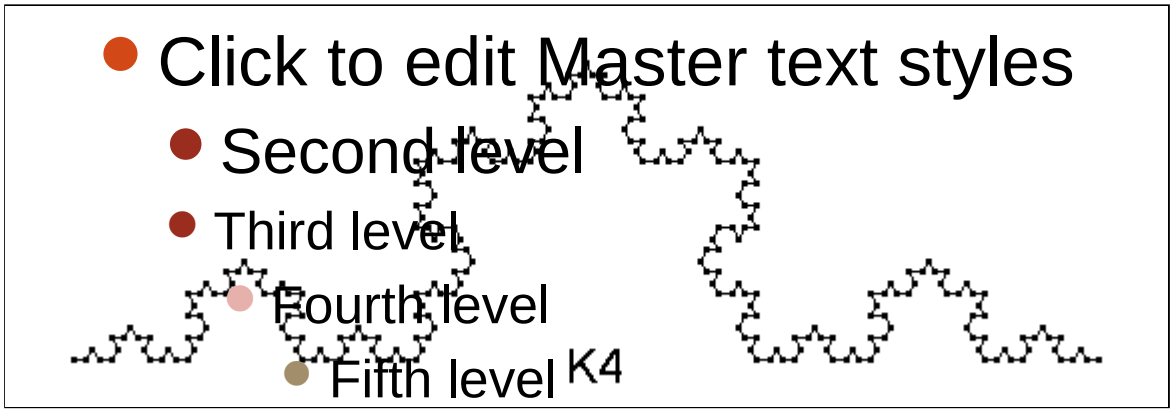


- In po 4 dneh



Končno ji je njena najboljša prijateljica, krožnica, povedala za čudovito kremo! Če se namaže z njo po vsakem delčku svoje kože, bo rast mozoljev vsaj ustavljena. A krema je draga ... In za vsak cm potrebuje črta vsaj 6g te kreme. Koliko jo mora kupiti, če je bila na začetku dolga *d* in je preteklo že *n* dni, kar je dobivala mozolje?

Neprijeten pogled na črtino kožo



Ideja

- Črta je po 4 dneh taka, kot bi vzeli 4 take črte med A' in B', kakršne so bile po treh dneh.
- A' in B' pa sta na tretjini razdalje med A in B

Ven z 'a'-ji

- Iz niza sestavi nov niz (ali lahko spremenimo obstoječega?), ki pa ne vsebuje 'a' jev
- `def odstraniA(niz):`
- `    novi = ""`
- `    for znak in niz:`
- `        if znak != "a":`
- `            novi += znak`
- `    return novi`

Rekurzivno

```
def odstraniA(niz):
    if niz == "":
        return ""
    prvi = niz[0]
    preostanek = niz[1:]
    novi = odstraniA(preostanek)
    if prvi != "a":
        return prvi + novi
    return novi
```

Seznam besed v nizu

- 'Matija Mataja hruške prodaja'
  - ['Matija', 'Mataja', 'hruške', 'prodaja']
- `>>> niz = 'Matija Mataja hruške prodaja'`
- `>>> sez = niz.split(' ')`
- `>>> sez`
- `['Matija', 'Mataja', 'hruške', 'prodaja']`
- `>>> niz = 'Matija  Mataja  hruške prodaja'`
- `>>> sez = niz.split(' ')`
- `>>> sez`
- `['Matija', '', '', 'Mataja', '', '', '', '', '', '', 'hruške', 'prodaja']`
- `>>>`

Malo čaranja

- `def seznamBesed(niz):`
- `    return [i for i in niz.split(" ") if i != ""]`
- Seznamski izrazi
  - `[izraz seznam pogoj]`
- Seveda znamo tudi brez čaranja
- Recimo kar
- `def seznamBesed(niz):`
- `    return niz.split()`

Kaj pa z rekurzijo?

- Brez split seveda!
- Postopek
  - Določi prvo besedo bes,
  - na preostanku niza niz rekurzivno izračunaj seznam besed sez,
  - prvo besedo dodaj na začetek seznama sez in vrni tako popravljen seznam sez.

Zaustavitveni pogoj

- Več jih je!
- Ko v obdelavo dobimo
  - niz z eno besedo
  - [beseda]
  - niz sestavljen iz samih presledkov
  - []
  - prazen niz
  - []
- Kako prepoznamo primere:
  - preostanek je prazen niz
  - poskušamo pridobiti prvo besedo, a neuspešno
  - == ''

Koda

```
def seznamBesed(niz):
    if niz == "":
        return []
    bes, preostanek = razdeliNaPrviBesedi(niz)
    if bes == "":
        return []
    if preostanek == "":
        return [bes]
    sez = seznamBesed(preostanek)
    sez.insert(0, bes)
    return sez
```

razdeliNaPrviBesedi

- Vračamo dva podatka!
- Če niza "ni", vrnemo dva prazna niza (besedo in preostanek)
- Če je prvi znak presledek
  - Rezultat je tak, kot bi uporabili to funkcijo na nizu brez prvega (rekurzivni klic)
- Če pa prvi ni presledek
  - Poiščemo mesto, kje nastopa prvič
  - Find
  - Če vrne < 0 – presledka ni
  - Če pa je, le razrežemo niz

Koda

```
def razdeliNaPrviBesedi(niz):
    if niz == "":
        return "", ""
    if niz[0] == " ":
        return razdeliNaPrviBesedi(niz[1:])
    ind = niz.find(" ")
    if ind < 0:
        return niz, ""
    return niz[:ind], niz[ind + 1:]
```

Obrni niz

- Napiši funkcijo, ki s pomočjo rekurzije vrne obrnjeni niz.
- Ideja:
  - Matija – M + atija
  - Obrnjeni niz dobimo tako, da M dodamo na konec obrnjenega niza atija!
  - In kako dobimo obrnjeni niz 'atija' – z istim postopkom!
  - Ustavitveni pogoj?
- Ideja 2:
  - Obrnjeni niz dobimo tako, da a dodamo na začetek obrnjenega niza 'Matij'
- Ideja 3:
  - Niz razdelimo na pol, obrnemo vsako polovico in staknemo dobljena niza

Palindrom

- S pomočjo rekurzije preveri, če je niz palindrom.
- Niz je palindom, če velja:
  - Prvi znak je enak zadnjemu
  - Vsi znaki brez prvega in zadnjega so tudi palindromski niz
- Različica naloge:
  - Dan je seznam. S pomočjo rekurzije preveri, če je "palindromski". Npr. [1, 3, 3, 1] je palindromski seznam
- Različica naloge:
  - Dan je seznam nizov. Preveri, če je dvojno palindromski. To pomeni, da je kot seznam palindromski in da so tudi nizi, ki ga sestavljajo, palindromi.
- Različica naloge:
  - Dan je seznam števil. Preveri, če je dvojno palindromski. To pomeni, da je kot seznam palindromski in da so tudi števila, ki ga sestavljajo, palindromska. Število je palindomsko, če se z leve bere enako kot z desne.

Urejanje z zlivanjem

- Seznam števil lahko uredimo po naslednjem postopku
- Razdelimo ga na pol na dva seznama.
- Oba dobljena seznama uredimo z enakim postopkom (rekurzivna klica).
- Dobljena urejena seznama združimo v novega, prav tako urejenega s postopkom, ki mu rečemo, zlivanje.

Zlivanje – različice besedila naloge

- Dan je nek niz. Naredi nov seznam, ki vsebuje enake znake kot prvotni niz, le da so urejeni po abecedi.
- Namesto, da urejaš od najmanjšega do največjega, uredi od največjega do najmanjšega (bodisi seznam, bodisi niz).
- Dan je seznam seznamov. Uredi ga tako, da bodo seznam urejen glede na dolžino seznamov, ki nastopajo v njem.
- Dan je seznam nizov. Uredi ta seznam glede na dolžino nizov.
- Dan je seznam seznamov števil. Uredi ga tako, da bodo seznam urejen po dolžini seznamov, "notranji" seznami pa bodo imeli elemente urejene po velikosti.
- Dan je seznam seznamov števil. Uredi ga tako, da bodo seznam urejen tako,da bodo najprej tisti seznami, ki imajo najmanjše elemente, "notranji" seznami pa bodo imeli elemente urejene po velikosti.
- Dan je seznam seznamov števil. Uredi ga tako, da bodo "notranji" seznami imeli elemente urejene po velikosti, celotni seznam pa bo urejen "leksikografsko". Torej najprej bodo tisti seznami, ki imajo najmanjše elemente, če pa imata dva seznama enak najmanjši element, primerjamo naslednja dva najmanjša elementa teh dveh seznamov, …

Belokranjski vzorci I

- Belokranjski vzorci za vezenine stopnje 1, 2, in 3 so naslednji:
- 
-
- Npr. vzorec stopnje 3 dobimo tako, da poln kvadrat s stranico a razdelimo na 9 enakih kvadratov. Izrežemo 4 stranske srednje kvadrate. Nato postopek 2x ponovimo na preostalih 5 kvadratih.
- Sestavite funkcijo, ki izračuna, kolikšna je dolžina niti, ki jo potrebujemo za izdelavo vzorca stopnje n, če za najmanjše vbode porabimo 3mm niti (za vzorec stopnje 1 torej porabimo 5 x 4 × 3mm = 60mm).