

# Uvod u programiranje



2023/24



# Rekurzija



# Rekurzija

- U matematici ili informatici označava postupak ili funkciju koji u svojoj definiciji koriste sami sebe
- Rekurzivne definicije su prisutne i u matematici:

## Definicija prirodnih brojeva:

- 1 je prirodan broj
- Ako je  $n$  prirodan broj, onda je to i  $n+1$

## Fibonačijev niz:

- 1. član niza je 0
- 2. član niza je 1
- Svaki  $n$ -ti član niza ( $n > 2$ ) je suma prethodna dva ( $a_n = a_{n-1} + a_{n-2}$ )

# Rekurzija

- U matematičkom smislu rekurzija predstavlja definisanje problema uz pomoć samog tog problema.
- Drugi način definisanja rekurzije kaže da rekurzija predstavlja način definisanja problema preko pojednostavljene verzije istog tog problema.
- Jedan primer kojim se ovo može opisati je rešavanje problema pronalaska puta do kuće (problem ćemo označiti izrazom “pronađi put do kuće”). Rekurzijom bi se ovaj problem mogao opsati u tri koraka, na sledeći način:
  - 1) ako si kod kuće, ostani u mestu,
  - 2) inače, napravi jedan korak prema kući,
  - 3) pronađi put do kuće.
- Tačka pod brojem 3 u ovom opisu problema predstavlja poziv istog problema iz definicije, ali posle učinjene jedne jednostavne akcije, a to je jedan korak prema kući, problem je pojednostavljen.

# Rekurzija

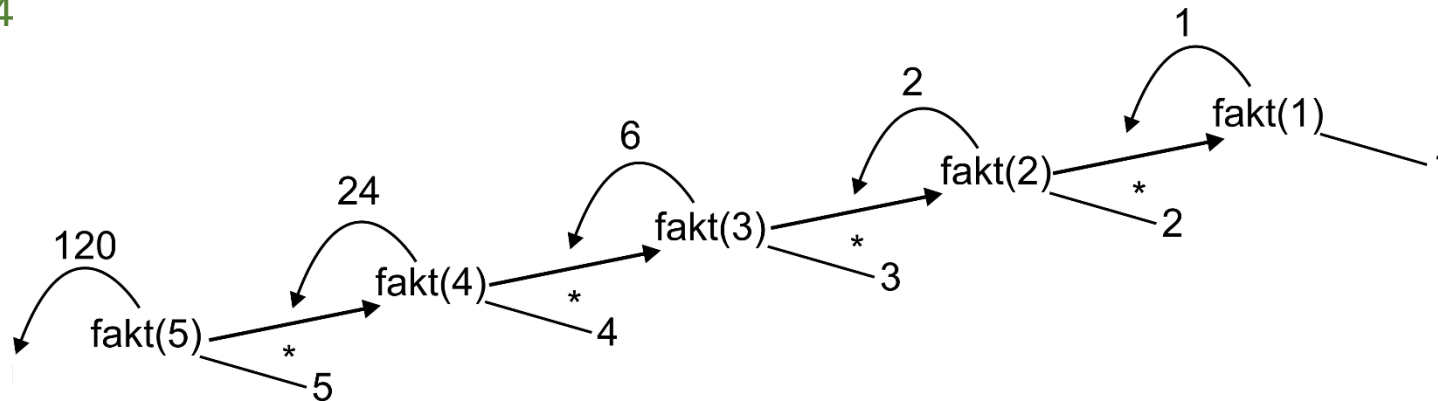
- Opisani postupak se može uopštiti, čime dobijamo generalni algoritam koji rešava probleme rekurzijom, i on se sastoji od sledeća tri koraka:
  - 1) trivajlni slučaj (kojim se prekida proces izračunavanja),
  - 2) izvršavanje jedne akcije koja nas vodi ka trivijalnom slučaju,
  - 3) rekurzivni poziv.
- Ovako opisani postupak rešavanja problema ima algoritamski oblik i za većinu problema se može implementirati.

# Rekurzija - Faktorijel

```
def fakt(n):  
    if n<2:  
        return 1  
    else:  
        return n*fakt(n-1)  
print(fakt(5))
```

- $0! = 1$      $n! = n \cdot (n - 1)!$

```
fakt(5) = 5 · fakt(4)  
        = 5 · (4 · fakt(3))           // (jer je fakt(4) = 4 · fakt(3))  
        = 5 · (4 · (3 · fakt(2)))     // (jer je fakt(3) = 3 · fakt(2))  
        = 5 · (4 · (3 · (2 · fakt(1)))) // (jer je fakt(2) = 2 · fakt(1))  
        = 5 · (4 · (3 · (2 · 1)))     // (jer je fakt(1) = 1)  
  
        = 5 · (4 · (3 · (2 · 1)))     // množenje se sada vrši redom  
        = 5 · (4 · (3 · 2))           // kojim se f-je završavaju, tj. unazad  
        = 5 · (4 · 6)  
        = 5 · 24  
        = 120
```



# Rekurzija

- Svako rešenje rekurzivnog problema se sastoji od dva glavna dela:
  - **Bazni slučaj(evi)** - rešenja trivijalnih instanci problema
  - **Rekurzivni slučajevi** - zavisnost rešenja problema od rešenja manjih instanci

**Definicija** Rekurzija predstavlja metod u kome rešenje polaznog problema nalazimo koristeći rešenja podproblema iste strukture - jednostavnije instance istog problema.

# Rekurzija – često postavljana pitanja

- Funkcija nije završila sa radom i ja ne mogu ponovo da je pozovem?
- Ukoliko funkcija pozove samu sebe, dobiću beskonačnu petlju?
- Kako da odredim koji su mi bazni slučajevi potrebni ili mi možda neki nedostaje?



# Rekurzija - Fibonači

```
def fibonacci (n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)  
  
print(fibonacci(5))
```

5

Fibonačijev niz:

1. član niza je 0

2. član niza je 1

Svaki n-ti član niza ( $n > 2$ ) je suma prethodna dva

( $a_n = a_{n-1} + a_{n-2}$ )

# Primer

- Napisati rekurzivnu funkciju kojom se cifre celog broja okreću u obrnutom redosledu.

```
def okreni_broj(n, reversed_n=0):  
    if n == 0:  
        return reversed_n  
    else:  
        zadnja_cifra = n % 10  
        preostali_broj = n // 10  
        return okreni_broj(preostali_broj, reversed_n * 10  
+ zadnja_cifra)  
  
# Primer poziva funkcije  
n = 12345  
print(okreni_broj(n))
```

# Verižni razlomci

- Verižni razlomak je izraz oblika:

$$\alpha = a_0 + \frac{b_0}{a_1 + \frac{b_1}{a_2 + \frac{b_2}{a_3 + \ddots}}}$$

gde su  $a_i, b_i$  proizvoljni izrazi.

Jednostavni verižni razlomak je verižni razlomak kojem su svi  $b_i$  jednaki 1.

# Verižni razlomci

- Jedan od verižnih razlomaka se vezuje za vrednost broja pi:

$$\frac{4}{\pi} = 1 + \frac{1}{3 + \frac{4}{5 + \frac{9}{7 + \frac{16}{9 + \frac{25}{11 + \frac{36}{13 + \dots}}}}}}$$

- Uočavamo dva niza:
  - Sabirci: 1, 3, 5, 7, 9, 11,...
  - Deljenici: 1, 4, 9, 16, 25, 36,...
- Rekurzija se ne može širiti u beskonačnost

# Verižni razlomci

```
import math
def verizni(n):
    if n>200:
        return 2*n-1
    else:
        return (2*n-1)+n*n/verizni(n+1)
```

```
print(verizni(1))
print(4/math.pi)
```

1.2732395447351628

1.2732395447351628