

# Algoritmi i strukture podataka

02 Rekurzija

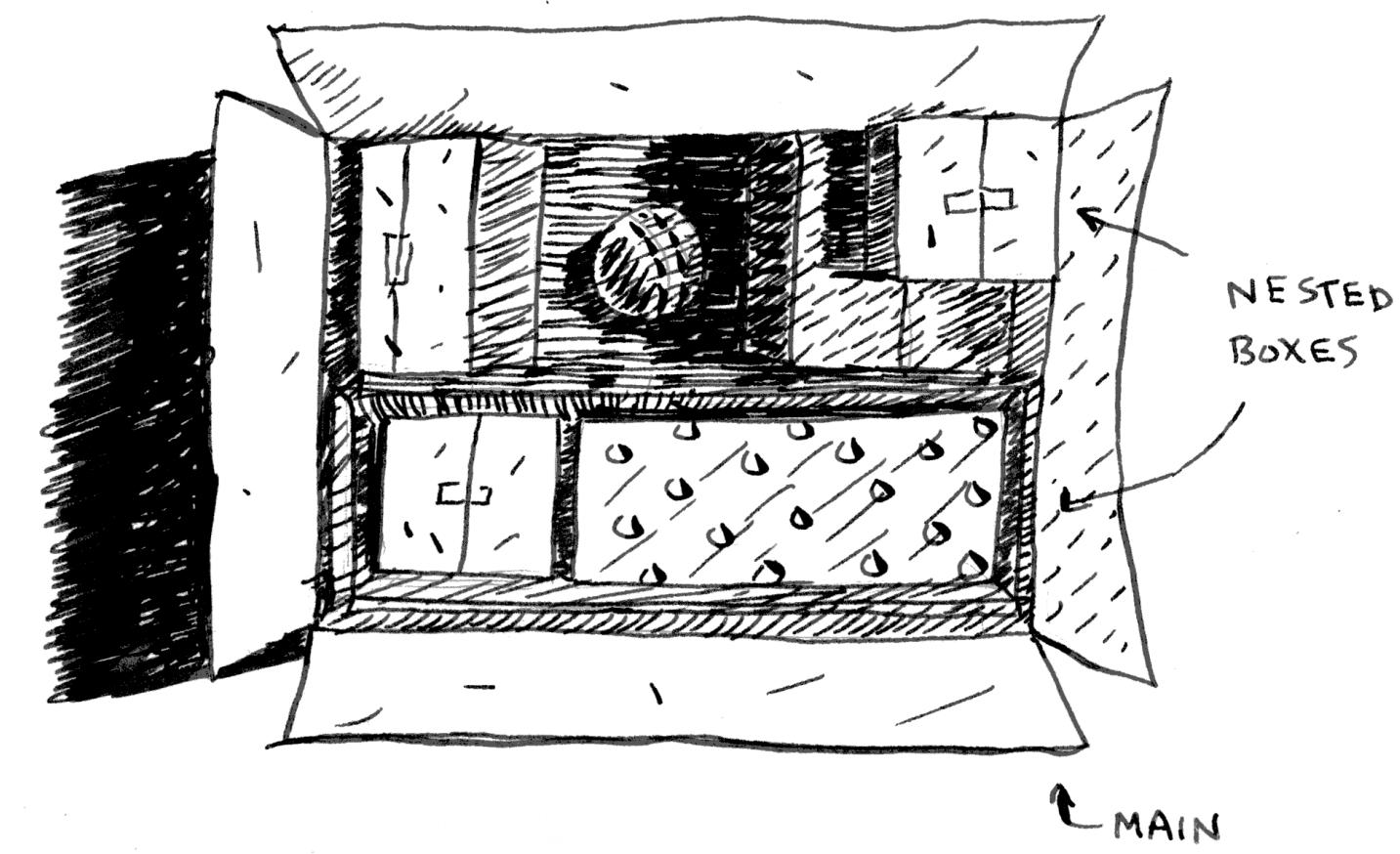
Katedra za informatiku, Fakultet tehničkih nauka, Novi Sad

2019

A screenshot of a Google search results page for the query "recursion". The search bar at the top contains "recursion". Below the search bar, there are several navigation links: "Све" (selected), "Слике", "Видео", "Књиге", "Вести", "Још", "Подешавања", and "Алатке". A search button with a microphone icon is also present. The main content area shows the following results:

- Да ли сте мислили: [recursion](#)**
- Recursion - Wikipedia**  
<https://en.wikipedia.org/wiki/Recursion> ▾ [Преведи ову страницу](#)  
Recursion occurs when a thing is defined in terms of itself or of its type. Recursion is used in a variety of disciplines ranging from linguistics to logic. The most ...  
[Recursion \(computer science\)](#) · [Recursion](#) · [Recursion \(disambiguation\)](#)
- Recursion (computer science) - Wikipedia**  
[https://en.wikipedia.org/wiki/Recursion\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Recursion_(computer_science)) ▾ [Преведи ову страницу](#)  
Recursion in computer science is a method of solving a problem where the solution depends on solutions to smaller instances of the  
[Types of recursion](#) · [Recursive program](#)

*"In order to understand recursion,  
one must first understand recursion."*



Preuzeto sa: <https://medium.freecodecamp.org/how-recursion-works-explained-with-flowcharts-and-a-video-de61f40cb7f9>

# Definicija

- Funkcija poziva samu sebe
- Sadržaj:
  - Osnovni slučajevi
    - Nema rekurzivnih poziva
    - Moraju postojati
  - Rekurzivni pozivi
    - Napredovanje ka osnovnom slučaju

# Tipovi

- Linearna
  - 1 rekurzivni poziv
  - Repna rekurzija
    - Poslednji korak je rekurzivni poziv
- Binarna
  - 2 rekurzivna poziva za svaki bazni slučaj
- Višestruka

# Rekurzija - poređenje

Dobre strane	Loše strane
Kraće	Teža za razumevanje kod nekih algoritama
Jednostavnije	Veće zauzeće memorije (stack overflow)
Čitljivije	Sporije od iterativnih postupaka (zbog mnogobrojnih poziva funkcije)
Elegantnije	Prekidanje postupka u toku izvršavanja
Neki algoritmi i strukture podataka se rekurzivno definiju pa ih je lakše rekurzivno implementirati	Zavisnost od broja poziva

# Zadatak 1

- Naći sumu prvih  $n$  prirodnih brojeva rekurzivnim putem.

## Zadatak 2

- Naći maksimalni element niza rekurzivnim putem.

# Zadatak 3

- Izračunati n-ti Fibonačijev broj pomoću:
  1. Linearne
  2. Binarne rekurzije.

\* Uporedite vreme izvršavanja za obe implementacije.

## Zadatak 4

- Naći sumu elemenata liste upotrebom:
  1. Linearne rekurzije
  2. Binarne rekurzije
- Analizirati maksimalnu dužinu liste za koje su postupci 1. i 2. upotrebljivi.

## Zadatak 5

- Napisati rekurzivnu funkciju za replikaciju. Funkcija ima dva parametra – prvi je broj koji treba umnožiti, a drugi broj ponavljanja.
- Primer poziva: `replicate(5,6)`
- Rezultat: `[5, 5, 5, 5, 5, 5]`

## Zadatak 6

- Prikazati absolutne putanje svih datoteka na zadatoj putanji u direktorijumskom stablu.
- *Napomena:* koristiti moduo **os**

## Zadatak 7

- Napisati program za konverziju pozitivnog celog broja zapisanog u dekadnom sistemu u string u zadatoj brojnoj osnovi.
- *Napomena:* brojne osnove mogu imati vrednost od 2 do 16

## Zadatak 8

- Tower of Hanoi
- Data je platforma sa tri stuba označena sa a, b i c. Na stubu a nalaze se n diskova složenih po veličini od najmanjeg ka najvećem. Napisati algoritam za prebacivanje diskova sa stuba a na stub b.
- *Napomene:*
  - Prebacuje se jedan disk po potezu.
  - Diskovi uvek moraju biti sortirani od najmanjeg ka najvećem.