Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Ivana Oreški

REKURZIJE

Završni rad

Osijek, 2011.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike

Ivana Oreški

REKURZIJE

Završni rad

Voditelj: prof. dr. sc. A. Klobučar

Osijek, 2011.

Sažetak. Rekurzija u matematici i računarstvu je metoda definiranja funkcija u kojima se definirajuća funkcija primjenjuje unutar definicije. U općem slučaju rekurzivni problemi definirani su u dva oblika: prvi, jednostavni oblik, koji predstavlja trivijalni slučaj, obično je zadan definicijom; drugi oblik gdje je problem definiran pomoću sličnoga (jednostavnijeg) problema. U matematici, rekurzivne relacije (rekurzije) su formule kod kojih se n-ti član nekog niza a_n izražava pomoću nekoliko prethodnih članova a_k, $k < n$. Precizna matematička podloga za taj pojam je princip definicije indukcijom. U ovom završnom radu prikazani su različiti tipovi rekurzivnih relacija, metode njihovog rješavanja te primjena na probleme prebrojavanja. Poznati rekurzivni problemi koji su obrađeni su hanojske kule, faktorijeli, Fibonaccijevi brojevi, Catalanovi brojevi i mnogi drugi. Razlikujemo linearne i nelinearne rekurzije. Linearne rekurzije s konstantnim koeficijentima dijelimo na homogene i nehomogene. Linearne homogene rekurzivne relacije s konstantim koeficijentima dijelimo na slučaj r različitih korijena karakteristične jednadžbe i slučaj kada postoje višestruki korijeni karakteristične jednadžbe. Veliki broj programskih zadataka predstavljaju rekurzivne probleme. Rekurzija nastaje kada funkcija (procedura) poziva samu sebe direktno ili indirektno.Vrlo često se u analizi nekih (rekurzivnih) algoritama javljaju algoritmi tipa "podijeli pa vladaj", oni rekurzivno razbijaju problem na dva ili više podproblema istog tipa, sve dok ovi ne postanu tako jednostavni da se mogu riješiti direktno. Rekurzivno definirani matematički objekti su također i fraktali koji pokazuju sličnost samome sebi.

Ključne riječi: rekurzije (rekurzivne relacije), matematička indukcija, rekurzivni problemi, sličnost, karakteristična jednadžba, linearne homogene rekurzije s konstantnim koeficijentima - različiti i višestruki korijeni, linearne nehomogene rekurzije s konstantnim koeficijentima, nelinearne rekurzije, sustavi rekurzija, metoda "podijeli pa vladaj".

Abstract.(Recursion) In mathematics and computer science, recursion is a method of defining functions in which the defining function is applied within the definition. In the general case, recursive problems are defined in two forms: first, simple form, which is a trivial case, it is usually given by definition; the second form, where the problem is defined using a similar (simpler) problem. In mathematics, the recursive relation (recursion) are formulas in which the n-th member of a sequence a_n is expressed by several members of the previous a_k, $k < n$. The precise mathematical foundation for this concept is the principle of definition by induction. In this final paper are shown different types of recursive relationships, methods of their solution and application to counting problems. Known recursive problems that have surfaced are Towers of Hanoi, the factorial, Fibonacci numbers, Catalano numbers and many others. We distinguish between linear and nonlinear recursion. Linear recursion with constant coefficients can be divided into homogeneous and inhomogeneous. Homogeneous linear recursive relation with constant coefficients are divided on the case r different roots of the characteristic equation and the case when there are multiple roots of characteristic equation. A large number of programming tasks are recursive problems.

Recursion occurs when a function (procedure) calls itself directly or indirectly. Often in the analysis of some (recursive) algorithm algorithms “ divide and conquer ” are reported, they recursively breaking the problem into two or more subproblems of the same type, as long as these do become so easy that it can be solved directly. Recursively defined mathematical objects are also fractals, which shows similarities to himself.

Keywords: recursion (recursive relations), mathematical induction, recursive problems, similarity, characteristic equation, linear homogeneous recursion with constant coefficients - various and multiple roots, non-homogeneous linear recursion with constant coefficients, nonlinear recursion, recursion systems, methods of “ divide and conquer ”.

Sadržaj

1. Uvod **1**

2. Općenito o rekurzijama **2**

2.1. Rekurzije i matematička indukcija 2

2.2. Jednostavni primjeri rekurzija 4

2.2.1. Hanojske kule 6

2.2.2. Fibonaccijevi brojevi 8

3. Linearne rekurzije **11**

3.1. Linearne homogene rekurzije s konstantnim koeficijentima 11

3.1.1. Postupak za rješavanje linearnih homogenih rekurzija s konstantnim koeficijentima 11

3.1.2. Različiti korijeni 12

3.1.3. Višestruki korijeni 13

3.2. Linearne nehomogene rekurzije s konstantnim koeficijentima 16

3.2.1. Postupak za rješavanje linearnih nehomogenih rekurzija s konstantnim koeficijentima 16

4. Sustavi rekurzija i nelinearne rekurzije **18**

5. Rekurzivne relacije za neke kombinatorne brojeve **21**

5.1. Catalanovi brojevi 21

5.1.1. Problem triangulacije konveksnog n-terokuta 21

5.1.2. Problem zagrada 22

5.2. Bellov i Stirlingov broj 23

5.3. Dvoindeksne rekurzije 24

6. Rekurzije u programiranju **25**

6.1. Podijeli pa vladaj (Devide and Conquer) 28

6.1.1. Merge Sort (sortiranje sažimanjem) 29

6.1.2. Quick Sort (brzo sortiranje) 29

7. Rekurzija u slikama **30**

Literatura **32**

1. Uvod

Cilj ovog rada je upoznavanje s pojmovima i objektima koji su predmet proučavanja mnogih matematičara, od kojih je centralan pojam rekurzija. Metoda rješavanja mnogih praktičnih problema je svođenje problema na jednostavniji slučaj. Dalje se i taj jednostavniji slučaj na sličan način dodatno može pojednostavniti sve dok ne dođemo do osnovnih problema čije je rješenje očigledno. Takav pristup rješavanju problema naziva se "rekurzija", a problemi na koje je ovaj pristup primjenjiv označavaju se kao rekurzivni problemi.

Kroz drugo poglavlje obradit će se osnovni pojmovi vezani uz rekurziju i ukratko će se opisati neke poznate rekurzije kao što su definicija faktorijela i suma prirodnih brojeva. Još jedan primjer rekurzije je problem "kule Hanoja", a osnova problema su tri štapa. Spomenut će se matematička indukcija kao precizna matematička podloga za pojam rekurzivnih relacija.

Nakon što smo naveli najbitnije o rekurzijama, dolazimo do najvažnijih poglavlja, a to je ono o vrstama rekurzivnih relacija. Tipične rekurzivne relacije kojima ćemo se ovdje baviti izgledat će ovako: $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r}$, gdje su c_i neke (zadane) konstante, te $r \in \mathbb{N}$ dani čvrst broj, zatim $a_n = c a_{n-1} + f(n)$, gdje je c zadana konstanta, a $f(n)$ zadana funkcija. U glavnom dijelu rada bit će obrađene vrste rekurzivnih relacija, linearne i nelinearne. Najprije će se obraditi linearne homogene rekurzivne relacije s konstantnim koficijentima, tu razlikujemo dva slučaja, slučaj kada imamo r različitih korijena karakteristične jednadžbe i slučaj kada postoje višestruki korijeni karakteristične jednadžbe. Navest ćemo rekurzivnu definiciju tzv. Fibonaccijevog niza brojeva F_n koja je linearna homogena rekurzivna relacija s konstantnim koeficijentima. Zatim će se obraditi linearne nehomogene rekurzivne relacije sa konstantnim koeficijentima, dat će se mnogi primjeri i postupci vezani uz rješavanje tih problema.

Zatim u petom poglavlju će biti navedeni razni tipovi rekurzivnih relacija, među njima i relacije sa nekonstantnim koeficijentima. Također će biti govora i o nekim poznatim kombinatornim problemima.

Na kraju rada ukratko će se opisati gdje se, osim u matematici, pojavljuju rekurzije. Navest će se neke primjene u računarstvu i u slikama. Općenitije se naziv rekurzija koristi za opis procesa ponavljanja objekata na samosličan način. Primjerice, kada su površine dvaju zrcala gotovo uzajamno paralelne, ugniježđene slike koje se pojavljuju su oblika rekurzije. Svrha ovog rada je poučiti sve čitatelje da nauče i da dobiju predodžbu kako razmišljati u rekurzivnom stilu, rekurzivno programirati, korektno upotrebljavati rekurzije u raznim primjerima i prepoznati rekurzije u prirodi i raznim slikama.

2. Općenito o rekurzijama

Rekurzija je u matematici i računarstvu metoda definiranja funkcija u kojima se definirajuća funkcija primjenjuje unutar definicije. Rekurzija konstruira klasu objekata ili metoda (ili objekata iz određene klase) definiranjem nekoliko jednostavnih osnovnih slučajeva ili metoda (često samo jednu), i potom definiranjem pravila za razbijanje složenih slučajeva u jednostavnije. Primjerice, rekurzivna definicija predaka osobe je:
• Nečiji roditelji su njegovi pretci (*osnovni slučaj*);
• Roditelji bilo kojeg pretka su također pretci osobe koju promatramo (*korak rekurzije*).
Zgodno je zamisliti da rekurzivna definicija definira objekte u terminima "prethodno definiranih" objekata definirajuće klase. Definicije poput ove su česte u matematici.

Neka je definiran niz brojeva (a_n). Tada **rekurzivnom relacijom(rekurzivnom formulom ili rekurzijom)** zovemo svaku relaciju (često formulu), pomoću koje se n-ti član niza a_n izražava pomoću nekoliko prethodnih članova a_k, $k < n$. Rekurzivna relacija može elemente niza izražavati uvijek pomoću fiksnog broja prethodnika (za neki fiksni $k \in \mathbb{N}$ proizvoljni element niza a_n izražava pomoću $a_{n-1}, \ldots, a_{n-k}$) ili pomoću svih njegovih prethodnika. U prvom slučaju govorimo o rekurzivnim relacijama konačne, a u drugom o rekurzivnim relacijama beskonačne prošlosti. Potrebno je imati početne uvjete (vrijednosti elemenata niza za neke vrijednosti indeksa) da bi se rekurzivna relacija mogla koristiti za računanje elemenata niza. Najčešće su to prvi elementi niza. Za rekurzivne relacije konačne prošlosti obično uzimamo prvih k vrijednosti niza, npr. $a_1, a_2, \ldots, a_k$, a za beskonačne prošlosti prvi element niza. Početni uvjeti se ili zadaju ili su dostupni jednostavnim računom.

2.1. Rekurzije i matematička indukcija

Princip definicije indukcijom (ili **rekurzijom**) Neka je S skup, $a \in S$, te neka je za svako $n \in \mathbb{N}$ dano neko preslikavanje $f_n \colon S \times S \times \ldots \times S \to S$. Tada postoji jedinstveni niz (x_n) u S, takav da je

$$x_1 = a$$

$$x_{n+1} = f_n(x_1, x_2, \ldots, x_n), \text{ za svako } n \in \mathbb{N}. \tag{1}$$

Posebno, ako f_n ovisi samo o varijabli x_n, tj. ako je $f_n(x_1, x_2, \ldots, x_n) = f_n(x_n)$, onda zapravo imamo za svako $n \in \mathbb{N}$ preslikavanje $f_n \colon S \to S$, pa postoji jedinstven niz $x_1, x_2, \ldots, x_n, \ldots$ u S, takav da je $x_1 = a$ i $x_{n+1} = f_n(x_n)$, za svako $n \in \mathbb{N}$. Formula (1) se zove **rekurzivna formula** (ili **relacija**) za niz (x_n).

Definicija 2.1. *Neprazan skup $\mathbb{N}$ zove se skup prirodnih brojeva, a njegovi elementi prirodni brojevi ako vrijede sljedeći Peanovi aksiomi:*
P1) Postoji funkcija sljedbenika $s \colon \mathbb{N} \to \mathbb{N}$;
P2) Postoji barem jedan element $1 \in \mathbb{N}$ takav da je $s(n) \neq 1$, $\forall n \in \mathbb{N}$;
P3) Ako je $s(m) = s(n)$ za $m, n \in \mathbb{N}$ onda je $m = n$;

P4) Princip matematičke indukcije: Ako je $M \subseteq N$ i ako vrijedi

1) $1 \in \mathbb{M}$

2) $n \in \mathbb{M} \Rightarrow s(n) \in M, \forall n \in \mathbb{N},$

tada je $M = \mathbb{N}$.

Uočimo da su 1. i 2. Peanov aksiom rekurzivna definicija prirodnih brojeva. U četvrtom aksiomu govori se o prelasku sa m-tog slučaja u rješavanju nekog problema na $m+1$ slučaj, što podsjeća na naše rekurzivne definicije. Zato nije neočekivano da su matematička indukcija i rekurzivne relacije u bliskoj vezi koju ćemo pokazati na primjeru.

Primjer 2.1. *Neka je niz brojeva* A_n *definiran rekurzivno*

$$\begin{aligned} A_0 &= 100 \\ A_n &= 2 \cdot A_{n-1} \end{aligned}$$

Iterativnim postupkom možemo odrediti vrijednosti niza:

$$\begin{aligned} A_1 &= 2A_0 = 200 \\ A_2 &= 2A_1 = 400 \\ A_3 &= 2A_2 = 800 \\ A_4 &= 2A_3 = 1600 \\ &\vdots \end{aligned}$$

Za određivanje vrijednosti A_3 *potrebna su 3 množenja, za određivanje* A_4 *četiri množenja. Možemo pretpostaviti da je za određivanje vrijednosti* A_n *potrebno* n *množenja. Ta pretpostavka se može dokazati metodom matematičke indukcije.*
Možemo na osnovu prvih pet vrijednosti niza uočiti pravilo da je $A_n = 100 \cdot 2^n$. *Uočavamo da je ova relacija točna za* $n = 0, 1, 2, 3, 4$. *Provjerimo da li ona vrijedi za sve vrijednosti* n.

Kao metodu provjere naše pretpostavke koristit ćemo metodu matematičke indukcije:
1. korak (baza): Provjerimo da li tvrdnja vrijedi za $n = 0$. Trebamo dokazati da je $100 \cdot 2^0 = A_0$. Kako je $100 \cdot 2^0 = 100$ i $A_0 = 100$ (iz definicije niza A_n) zaključujemo da je baza indukcije točna.
2. korak (pretpostavka): Pretpostavimo da naša formula vrijedi u slučaju $n = m$ gdje je $m \geq 0$ prirodan broj, odnosno pretpostavimo da vrijedi

$$A_m = 100 \cdot 2^m \tag{2}$$

3. korak (korak indukcije): Dokažimo da formula vrijedi i u slučaju $n = m + 1$, odnosno dokažimo da

$$A_{m+1} = 100 \cdot 2^{m+1}$$

Počnimo od lijeve strane i primijenimo rekurzivnu definiciju niza A. Dobivamo:

$$A_{m+1} = 2 \cdot A_m$$

Prema indukcijskoj pretpostavci je $A_m = 100 \cdot 2^m$ pa je sada

$$A_{m+1} = 2 \cdot 100 \cdot 2^m = 100 \cdot 2 \cdot 2^m = 100 \cdot 2^{m+1}$$

što je i trebalo dokazati. Sada smo sigurni da (2) vrijedi za svako n. Ovakva forma definiranja niza brojeva naziva se definicija u zatvorenom obliku. Osnovna prednost je u tome što je broj računskih operacija potrebnih za izračunavanje vrijednosti niza mali. Npr. za $n = 100$ kod iterativnog pristupa imamo 100 množenja, kod rekurzivnog 100 množenja i 100 rekurzivnih poziva dok korištenjem (2) imamo samo dvije računske operacije.

2.2. Jednostavni primjeri rekurzija

Kao prvi primjer rekurzivne definicije navest ćemo **faktorijel** prirodnog broja n, u oznaci $n!$. Može se definirati kao $n! = 1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n$.
Alternativni način definiranja faktorijela je:

$$0! := 1$$

$$n! = (n-1)! \cdot n \tag{3}$$

gdje prvom relacijom definiramo najjednostavniji (trivijalni) slučaj, a drugom relacijom definiramo način prevođenja problema reda n na jednostavniji problem za jedan manjeg reda. Na taj način, ma koliko n bio velik nakon konačno mnogo koraka doći ćemo do trivijalnog slučaja i izračunati $n!$

Na primjeru 2.1. smo vidjeli da rekurzivno definiran problem može imati i nerekurzivni (direktni) način rješavanja, odnosno da se rješenje nekih rekurzivnih problema može pronaći u zatvorenoj formi. Promotrimo poznati problem određivanja sume prvih n prirodnih brojeva

$$S_n = 0 + 1 + 2 + \cdots + (n-1) + n$$

Ta definicija nije rekurzivna ali do rekurzivne definicije možemo doći na jednostavan način:

$$S_n = 0 + 1 + 2 + \cdots + (n-1) + n = [0 + 1 + 2 + \cdots + (n-1)] + n = S_{n-1} + n$$

Dakle, problem definiran rekurzivno je:

$$S_0 = 0, \quad S_n = S_{n-1} + n$$

Rješenje problema u zatvorenoj formi: $S_n = \frac{n(n+1)}{2}$.

Spomenimo i da je svaki član aritmetičkog niza[1], osim prvog jednak je aritmetičkoj sredini neposrednih susjeda: $a_{n+1} = \frac{a_n + a_{n+2}}{2}$. Ta jednakost se može zapisati u obliku $a_{n+2} = 2a_{n+1} - a_n$, time smo dobili rekurzivnu relaciju. Da bi aritmerički niz bio određen, u ovom slučaju potrebno je zadati njegova prva dva člana a_1 i a_2.

[1]Definicija: Neka su $a_1, d \in \mathbb{R}$. Niz realnih brojeva (a_n) čiji je opći član zadan s $a_n = a_1 + (n-1)d$, $n \in \mathbb{N}$ nazivamo aritmetički niz s diferencijom d.

Svaki član geometrijskog niza[2] (izuzev prvog) jednak je geometrijskoj sredini svojih neposrednih susjeda. Za geometrijski niz imamo očito najjednostavniju rekurzivnu relaciju: $a_{n+1} = qa_n$

Primjer 2.2. *Nađite rekurziju za broj $P(n)$ permutacija[3] skupa $\mathbb{N}_n$ i izračunajte ga.*

Rješenje: Svaka permutacija $i_1, i_2, \ldots, i_n$ od $\mathbb{N}_n = 1, 2, \ldots, n$ dobiva se na jedinstven način iz permutacije elemenata skupa $\mathbb{N}_{n-1} = 1, 2, \ldots, n-1$ umetanjem elemenata n ili na prvo mjesto, ili između prvog i drugog mjesta itd., ili na zadnje $(n - to)$ mjesto. Stoga je za $n \geq 2$

$$P(n) = nP(n-1).$$

To zajedno s početnom vrijednosti $P(1) = 1$ daje

$$\begin{aligned} P(n) = nP(n-1) = n(n-1)P(n-2) = \ldots = n(n-1)(n-2)\ldots 2P(1) = \\ = n(n-1)(n-2)\ldots 2 \cdot 1 = n! \end{aligned}$$

Primjer 2.3 (**Podjela ravnine pravcima**). *Neka je u ravnini zadano n pravaca $p_1, p_2, \ldots, p_n$ u općem položaju (nema paralelnih i nikoja tri ne prolaze istom točkom) i neka je $d_2(n)$ broj dijelova na koje ti pravci dijele ravninu, $n \in \mathbb{N}_0$.*
a) Nađite rekurzivnu relaciju među brojevima $d_2(n)$,
b) Nađite opći član $d_2(n)$.

Rješenje: a) Da bismo odredili broj dijelova $d_2(n)$ promatrajmo broj dijelova $d_2(n-1)$ na koje $n-1$ pravac ravnine $p_1, p_2, \ldots, p_{n-1}$ dijeli ravninu, i n-ti pravac p_n. Dodavanjem n-tog pravca broj dijelova $d_2(n-1)$ povećava se za n (to je broj dijelova što ga $n-1$ pravac određuje na n-tom pravcu). Prema tome, vrijedi rekurzivna relacija

$$d_2(n) = d_2(n-1) + n.$$

b) Da se nađe eksplicitni izraz za $d_2(n)$ iteriramo tu rekurzivnu relaciju, tj. pišemo umjesto n redom $n-1, n-2, \ldots, 1$:

$$\begin{aligned} d_2(n-1) &= d_2(n-2) + n - 1 \\ d_2(n-2) &= d_2(n-3) + n - 2 \\ &\vdots \\ d_2(3) &= d_2(2) + 3 \\ d_2(2) &= d_2(1) + 2 \\ d_2(1) &= d_2(0) + 1 \end{aligned}$$

[2]Definicija: Neka su $a_1, q \in \mathbb{R}$, $q \neq 0$. Niz realnih brojeva (a_n) čiji je opći član zadan s $a_n = a_1 \cdot q^{n-1}$, $n \in \mathbb{N}$ nazivamo geometrijski niz s kvocijentom q.

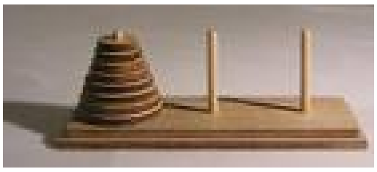
[3]Definicija:Neka je S skup od n elemenata, a $r \in \mathbb{N}$. Tada je r-permutacija skupa S uređene r-torka $(x_1, x_2, \ldots, x_r)$ kod koje su sve komponente $x_1, x_2, \ldots, x_r$ različiti elementi od S. Oznaka za broj svih r-permutacija skupa od n elemenata je $P(n, r)$.

Zbrajanjem svih formula[4] dobiva se

$$\begin{aligned} d_2(n) &= d_2(0) + n + (n-1) + (n-2) + \cdots + 2 + 1 = \\ &= 1 + \frac{n(n+1)}{2} = \binom{n}{0}\binom{n}{1}\binom{n}{2} = \frac{1}{2}(n^2 + n + 2) \end{aligned}$$

2.2.1. Hanojske kule

Još jedan primjer rekurzije je problem "**kule Hanoja**" koji je vjerojatno najpoznatiji rekurzivni problem u kompjuterskoj literaturi. Osnova problema su tri štapa A, B i C, koja su zabodena na ravnoj podlozi. Na jednom štapu, npr. na A, naslagano je n kolutova različitih polumjera poredanih po veličini tako da se najveći kolut nalazi na dnu, a najmanji na vrhu (v.sl. 2.1). Zadatak je prenijeti sve kolutove (jedan po jedan) s prvog na drugi štap. Postavlja se pitanje: Koliki je najmanji broj prijenosa a_n potreban da se svih n kolutova prenese s prvog na drugi štap?



Slika 2.1: Hanojske kule (tornjevi) (slika preuzeta sa [9])

Pri prebacivanju kolutova treba poštovati slijedeća pravila:

- U jednom potezu dozvoljeno je premjestiti samo jedan kolut.
- Nije dozvoljeno staviti veći kolut preko manjeg.
- Pri tome svaki od štapova možemo koristiti za privremeno smještanje kolutova. Može se npr. koristiti treći štap C za privremeni smještaj kolutova, ali uz poštovanje prethodna dva pravila.

Označimo sa a_n broj poteza potreban da se riješi zadatak sa n kolutova i promatrajmo jednostavne slučajeve.

Induktivni opis:

- Za $n = 1$ (jedan kolut) imamo samo jedan prijenos (tj. premještanje se može obaviti u jednom potezu) pa je $a_1 = 1$
- Pretpostavimo da znamo prenijeti n kolutova (imamo a_n prijenosa)

[4]Ta metoda se ponekad naziva i "teleskopiranje"

- Za prijenos $n+1$ koluta imamo sljedeće
  - Prenesemo n kolutova na drugi štap (ukupno a_n prijenosa);
  - Prenosimo najveći kolut na treći štap (ukupno jedan prijenos);
  - Prenesemo n kolutova s drugog na drugi štap (ukupno a_n prijenosa).

Dakle, vrijedi

$$a_{n+1} = 2a_n + 1, a_1 = 1$$

Rješenje je:

$$a_n = 2^n - 1, n \in \mathbb{N}$$

Napomena: Zapravo imamo $a_{n+1} \leq 2a_n + 1$. Ali budući da je $a_{n+1} \geq 2a_n + 1$ imamo jednakost.

Postavka problema ne uključuje rekurziju. U čemu je ovaj problem rekurzivan? Pokušajmo definirati temeljni slučaj i pravilo rekurzije.
Temeljni slučaj- Najednostavniji slučaj je ako štap sadrži samo jedan kolut, tada je rješenje jednostavno, prebaci se taj kolut na ciljni, tj. drugi štap.
Rekurzivno pravilo- Ako kula sadrži $n-1$ kolutova, pomicanje kolutova se može izvesti u tri koraka:
1. Pomakni gornji $n-1$ kolut na pomoćni štap C .
2. Preostali donji kolut sa prvog štapa A pomakni na ciljni štap B.
3. Zatim kulu od $n-1$ kolutova s pomoćnog štapa C prebaci na ciljni štap B.

Rješenje problema u rekurzivnom obliku:
Za $n = 2$ gornji kolut ćemo premjestiti na pomoćni štap, donji kolut na drugi štap i u posljednjem potezu kolut sa trećeg štapa premjestiti na srednji štap, tako da je $a_2 = 2a_1 + 1 = 3$. Ako promatramo n kolutova, problem možemo riješiti tako da prvo premjestimo $n-1$ kolutova na zadnji štap u a_{n-1} poteza, zatim najveći kolut premjestimo na drugi štap i na kraju $n-1$ kolutova sa zadnjeg štapa premjestimo na drugi štap u a_{n-1} poteza. Ukupan broj poteza je $a_n = 2a_{n-1} + 1$.

$$\begin{aligned} a_1 &= 1 \\ a_n &= 2a_{n-1} + 1 \end{aligned}$$

Elementi niza a_n:

$$\begin{aligned} a_1 &= 1 \\ a_2 &= 2a_1 + 1 = 3 \\ a_3 &= 2a_2 + 1 = 7 \\ a_4 &= 2a_3 + 1 = 15 \\ a_5 &= 2a_4 + 1 = 31 \end{aligned}$$

Uočavamo pravilo $a_n = 2^n - 1$.

2.2.2. Fibonaccijevi brojevi

Leonardo od Pise (1170-1250), poznat još kao Fibonacci[5], postavio je 1202.g. u svom radu "Liber abaci" tzv. "(nerealističan) problem zečeva".

Shema razmnožavanja zečeva je slijedeća:
Svaki par zec- zečica (starih barem 2 mjeseca) tijekom svakog slijedećeg mjeseca dobiju par mladih (zeca i zečicu). Pretpostavlja se da zečevi nikad ne umiru.
Pitanje: Ako smo na početku godine počeli s s jednim novorođenim parom, koliko će biti ukupno parova zečeva početkom slijedeće godine, odnosno općenito nakon n mjeseci?

Nakon prvog mjeseca će biti još uvijek jedan par zečeva, jer oni još nisu zreli za oplodnju. Nakon dva mjeseca imamo dva para. Nakon tri mjeseca imamo tri para (orginalni par i njihovi potomci nakon drugog i trećeg mjeseca).
Sa f_n označimo broj parova zec-zečica nakon n mjeseci, tj. tijekom $(n+1)$-og mjeseca. Prema pretpostavci je $f_0 = f_1 = 1$. Da bismo izračunali f_n treba broju parova f_{n-1} koji su živjeli prethodni mjesec dodati novorođene parove koji mogu doći samo od f_{n-2} parova živih prije dva mjeseca. Stoga je rješenje jednoznačno određeno nizom prirodnih brojeva $(f_n)_{n\geq 0}$ tako da za svaki $n \geq 2$ vrijedi:

$$f_n = f_{n-1} + f_{n-2}.$$

Brojeve f_n zovemo **Fibonaccijevim brojevima** (ime im je dao francuski matematičar Lucas). No, najčešće se koriste "pomaknuti" Fibonaccijevi brojevi, $F_n := f_{n-1}$. Oni zadovoljavaju istu rekurzivnu relaciju kao i brojevi f_n, razlika je samo u početnim uvjetima.

Fibonaccijev slijed (niz) (F_n) definira se početnim vrijednostima

$$\begin{aligned} F_0 &= 1 \\ F_1 &= 1 \end{aligned}$$

i rekurzivnom relacijom

$$F_n = F_{n-1} + F_{n-2}, \quad \text{za} \quad n \geq 2 \tag{4}$$

gdje je F_n ***n*-ti Fibonaccijev broj**.

Ta relacija predstavlja problem određivanja n-tog Fibonaccijevog broja koja se svodi na rješavanje jednostavnijih problema određivanja $(n-1)$ i $(n-2)$ Fibonaccijevog broja. Nakon dvije početne vrijedosti, svaki sljedeći broj je zbroj dvaju prethodnika: 2+3 dat će 5, 3+5 dat će 8, itd.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
F_n	0	1	1	2	3	5	8	13	21	34	55	89	144	233	...

[5]Ime dolazi od "filius Bonacci"- Bonaccijev sin

Dakle, rješenje problema kojeg je postavio Fibonacci glasi: nakon 12 mjeseci, odnosno tijekom 13. mjeseca biti će $F_{13} = 233$ parova zečeva.

Propozicija 2.1 (A. de Moivre). *Za Fibonaccijev slijed* (F_n) *vrijedi "zatvorena formula"*

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right], \qquad n = 0, 1, 2 \ldots \tag{5}$$

Dokaz: Ideja je dokaza da se rješenje Fibonaccijeve rekurzivne formule $F_n = F_{n-1} + F_{n-2}$, $n \geq 2$ (zanemarujući za trenutak početne vrijednosti F_0, F_1) potraži u obliku $F_n = q^n$. Uvrštavanjem dobivamo

$$q^n = q^{n-1} + q^{n-2}$$

$$q^{n-2}(q^2 - q - 1) = 0 \quad \text{za } n \geq 2.$$

Pretpostavimo $q \neq 0$, pa je $F_n = q^n$ rješenje Fibonaccijeve rekurzije ako i samo ako je $q^2 - q - 1 = 0$, tj. ako i samo ako je q rješenje ove kvadratne jednadžbe, odnosno

$$q_{1,2} = (1 \pm \sqrt{5})/2$$

Prema tome su

$$F_n^{(1)} = \left(\frac{1+\sqrt{5}}{2}\right)^n \quad \text{i} \quad F_n^{(2)} = \left(\frac{1-\sqrt{5}}{2}\right)^n$$

rješenja Fibonaccijeve rekurzivne formule. Tada vrijedi

$$F_n^{(1)} = F_{n-1}^{(1)} + F_{n-2}^{(1)}, \quad F_n^{(2)} = F_{n-1}^{(2)} + F_{n-2}^{(2)}.$$

Množenjem prve jednakosti sa λ, a druge s μ i njihovim zbrajanjem dobivamo da je i njihova linearna kombinacija

$$G_n = \lambda F_n^{(1)} + \mu F_n^{(n)}, \quad \lambda, \mu \in \mathbb{R}$$

rješenje, tj. $G_n = G_{n-1} + G_{n-2}$. Slijedi da je

$$F_n = \lambda\left(\frac{1+\sqrt{5}}{2}\right)^n + \mu\left(\frac{1-\sqrt{5}}{2}\right)^n$$

također rješenje rekurzije. Iz početnih vrijednosti za Fibonaccijeve brojeve $F_0 = 0, F_1 = 1$ dobivamo za

$$\begin{aligned} n &= 0: & \lambda + \mu = 0 \\ n &= 1: & \lambda\left(\frac{1+\sqrt{5}}{2}\right) + \mu\left(\frac{1-\sqrt{5}}{2}\right) = 1. \end{aligned}$$

Rješenje tog sustava linearnih jednadžbi je $\lambda = \frac{1}{\sqrt{5}}, \mu = \frac{-1}{\sqrt{5}}$, pa slijedi tvrdnja. ■

Napomena: Može se pokazati (iz zatvorene formule) da je

$$\lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2} \approx 1.618 = \phi. \tag{6}$$

Ovaj broj naziva se zlatni rez (božanski omjer). Uzmemo li jedan dio Fibonaccijevog niza, $2, 3, 5, 8,$ te podijelimo li svaki slijedeći broj s njemu prethodnim, dobit ćemo uvijek broj približan broju $1,618$ ($\frac{3}{2} = 1.5$, $\frac{5}{3} = 1.66$, $\frac{8}{5} = 1.6$). Odnosi mjera kod biljaka, životinja i ljudi sa velikom točnošću se približavaju broju ϕ. Primjerice, sjeme suncokreta raste u suprotnim spiralama, a broj ϕ daju međusobni odnosi promjera rotacije. Taj broj se dobiva i ako izmjerimo čovječju dužinu od vrha glave do poda, zatim to podijelimo s dužinom od pupka do poda.

Posljedica 1. *Broj F_n u Fibonaccijevom slijedu jednak je cijelom broju koji je najbliži broju*

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

Slijed (F_n) ima eksponencijalni rast.

3. Linearne rekurzije

3.1. Linearne homogene rekurzije s konstantnim koeficijentima

Rekurzivna relacija oblika

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r}, \quad n \geq r, \tag{7}$$

gdje su $c_i \in \mathbb{R}$ $(\mathbb{C})$, $i = 1, 2, \ldots, r$ zadane konstante i $c_r \neq 0$ naziva se **linearna homogena rekurzivna relacija s konstantnim koeficijentima reda** r za niz $(a_n)_{n\geq 0}$.

Pridjev "linearna" odnosi se na činjenicu da se javljaju samo prve potencije od a, dok pridjev "homogena" znači da nema konstantnih članova. Točnije, naziv "linearna homogena" dolazi od toga što skup svih rješenja od (7) čini linearni vektorski prostor (nad $\mathbb{R}$ ili $\mathbb{C}$). Odnosno, ako su $(a_n^{'})$ i $(a_n^{''})$ dva niza koja zadovoljavaju (7), onda uvrštavanjem $a_n^{'}$ u (7) i množenjem s λ, pa uvrštavanjem $a_n^{''}$ u (7) i množenjem s μ, te konačno zbrajanjem tih dviju jednakosti slijedi da i niz $(\lambda a_n^{'} + \mu a_n^{''})$, $\lambda, \mu \in \mathbb{C}$ zadovoljava (7). Imamo i "nul-rješenje" $a_n = 0$.

Jedan od najjednostavnijih primjera takvih rekurzija je Fibonaccijeva: $F_n = F_{n-1} + F_{n-2}$. Točnije, to je tročlana rekurzivna relacija s konstantnim koeficijentima. Npr. rekurzija $a_n = 2a_{n-1} + 4, n = 2, 3, \ldots$ nije homogena, a $a_n a_{n-1} + a_{n-2} a_{n-3} = 0, n = 3, 4 \ldots$ nije linearna. Rekurzivna relacija $a_n = (n+3)a_{n-1} + 4a_{n-2}$ nema konstantne koeficijente jer koeficijent $(n+3)$ uz a_{n-1} varira s n.

3.1.1. Postupak za rješavanje linearnih homogenih rekurzija s konstantnim koeficijentima

Postupak za rješavanje opće linearne rekurzije s konstantnim koeficijentima je sličan onom za Fibonaccijeve brojeve.
Rješenje jednadžbe (7)[6] se pretpostavlja u obliku linearne kombinacije partikularnih rješenja oblika $a_n = x^n$ (Eulerova supstitucija), za neko $x \neq 0$ (jer nas ne zanima"nul-rješenje" ili "trivijalno rješenje").
Uvrštavanjem $a_n = x^n$ u (7) dobiva se

$$x^n = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_r x^{n-r} / : x^{n-r} (x^{n-r} \neq 0)$$

$$x^r - c_1 x^{r-1} - c_2 x^{r-2} - \cdots - c_{r-1} x - c_r = 0 \tag{8}$$

Ta se jednadžba zove **karakteristična jednadžba** rekurzivne relacije (7). Po Osnovnom teoremu algebre ona ima u skupu kompleksnih brojeva r korijena $x_1, x_2, \ldots, x_r$ koji se zovu **karakteristični korijeni** od (7) (nisu nužno različiti, neki mogu biti međusobno jednaki). Zbog pretpostavke $c_r \neq 0$ niti jedan x_i nije 0.

[6]Postoji velika sličnost između ove teorije i teorije linearnih diferencijalnih jednadžbi s konstantnim koeficijentima. Ponekad se zbog toga rekurzivne relacije zovu i diferencijskim jednadžbama

Očito je da za $0 \neq x \in \mathbb{C}$, $a_n = x^n$ je rješenje od (7) ako i samo ako je x karakterističan korijen od (7).

Zbog činjenice da je skup svih rješenja od (7) vektorski prostor nad $\mathbb{C}$ proizlazi da su $x_1, x_2, \ldots, x_r$ karakteristični korijeni od (7), a $\lambda_1, \ldots, \lambda_r \in \mathbb{C}$, tada je i

$$a_n = \lambda_1 x_1^n + \lambda_2 x_2^n + \cdots + \lambda_r x_r^n \tag{9}$$

rješenje od (7).

Kažemo da je rješenje od (7) **opće rješenje** rekurzivne relacije (7) ako se svako rješenje može zapisati u obliku (9) za neki izbor konstanti $\lambda_1, \ldots, \lambda_r$.

Razlikujemo dva slučaja:

1. Slučaj r različitih korijena karakteristične jednadžbe.
2. Slučaj kada postoje višestruki korijeni karakteristične jednadžbe.

3.1.2. Različiti korijeni

Teorem 3.1. *Pretpostavimo da su svi karakteristični korijeni $x_1, x_2, \ldots, x_r$ rekurzivne relacije*

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r},\ c_r \neq 0,$$

$n \geq r$, međusobno različiti. Tada je opće rješenje dano sa

$$a_n = \lambda_1 x_1^n + \lambda_2 x_2^n + \ldots + \lambda_r x_r^n.$$

Dokaz: Neka je a_n neko rješenje te rekurzivne relacije. Tada je a_n potpuno određen početnim uvjetima $a_0 = b_0, a_1 = b_1, \ldots, a_{r-1} = b_{r-1}$. Treba pokazati da možemo odabrati konstante $\lambda_1, \lambda_2, \ldots, \lambda_r$, tako da vrijedi

$$\begin{gathered} \lambda_1 + \lambda_2 + \cdots + \lambda_r = b_0 \\ \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_r x_r = b_1 \\ \vdots \\ \lambda_1 x_1^{r-1} + \lambda_2 x_2^{r-1} + \cdots + \lambda_r x_r^{r-1} = b_{r-1} \end{gathered} \tag{10}$$

Sustav (10) je sustav od r linernih jednadžbi s r nepoznanica $\lambda_1, \lambda_2, \ldots, \lambda_r$. Matrica koeficijenata tog sustava je

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_r \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{r-1} & x_2^{r-1} & \ldots & x_r^{r-1} \end{bmatrix}$$

Ta je matrica poznata pod imenom **Vandermondeova matrica**. Njena determinanta je produkt

$$det\, V = \prod_{1\leq i<j\leq r} (x_j - x_i)$$

svih $\binom{r}{2}$ razlika oblika $x_j - x_i$. Kako su prema pretpostavci svi karakteristični korijeni $x_1, x_2, \ldots, x_r$ međusobno različiti, ta determinanta je različita od nule, pa sustav (10) ima jedinstveno rješenje po $\lambda_1, \lambda_2, \ldots, \lambda_r$. ■

Primjer 3.1. *Riješite rekurzivnu relaciju:* $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$*, uz početne uvjete* $a_1 = 1, a_2 = 2, a_3 = 3$.

Rješenje: Uvrštavanjem $a_n = x^n$ dobivamo karakterističnu jednadžbu

$$x^n - 2x^{n-1} - x^{n-2} + 2x^{n-3} = 0 \;/ : x^{n-3}$$

$$x^3 - 2x^2 + x - 2 = 0.$$

Izračunamo korijene: -1, 1, 2, pa je opće rješenje te rekurzije dano s

$$a_n = \lambda_1 \cdot (-1)^n + \lambda_2 \cdot 1^n + \lambda_3 \cdot 2^n.$$

Sada iz početnih uvjeta imamo

$$\begin{aligned} a_1 &= \lambda_1 \cdot (-1) + \lambda_2 + \lambda_3 \cdot 2 = 1 \\ a_2 &= \lambda_1 + \lambda_2 + \lambda_3 \cdot 4 = 2 \\ a_3 &= \lambda_1 \cdot (-1) + \lambda_2 + \lambda_3 \cdot 8 = 3 \end{aligned}$$

Jedinstveno rješenje tog sustava je $\lambda_1 = \frac{1}{6}$, $\lambda_2 = \frac{1}{2}$, $\lambda_3 = \frac{1}{3}$, pa je traženo rješenje

$$a_n = \frac{(-1)^n}{6} + \frac{1}{2} + \frac{2^n}{3}.$$

3.1.3. Višestruki korijeni

Lema 3.1. *Ako je kompleksni broj* x_0 *k-struki korijen polinoma* $P(x)$*, onda je on* $(k-1)$*-struki korijen derivacije* $P'(x)$.

Promatrajmo rekurziju

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r}, \;\; c_r \neq 0,$$

$n \geq r$, čija je karakteristična jednadžba $P(x) = x^r - c_1 x^{r-1} - c_2 x^{r-2} - \cdots - c_r = 0$.

Radi jednostavnosti pretpostavlja se da je x_0 trostruki korijen jednadžbe $P(x) = 0$, pa je $P(x) = (x - x_0)^3 Q(x)$, gdje je Q također polinom. Tada je za $n \geq r$, x_0 trostruki korijen polinoma $P_n(x) = x^{n-r}P(x) = x^n - c_1 x^{n-1} - \cdots - c_r x^{n-r}$.
Nadalje, x_0 je dvostruki korijen derivacije od $P_n(x)$:

$$P_n'(x) = nx^{n-1} - (n-1)c_1 x^{n-2} - (n-2)c_2 x^{n-3} - \cdots - (n-r)c_r x^{n-r-1},$$

pa stoga dvostruki korijen polinoma

$$xP_n'(x) = nx^n - (n-1)c_1 x^{n-1} - (n-2)c_2 x^{n-2} - \cdots - (n-r)c_r x^{n-r}.$$

Posebno je

$$n{x_0}^n = c_1 \cdot (n-1)x_0^{n-1} + c_2 \cdot (n-2)x_0^{n-2} + \cdots + c_r \cdot (n-r)x_0^{n-r},$$

pa iz toga slijedi da je $a_n = nx_0^n$ rješenje od (7). Nastavlja se postupak, dobiva se da je $a_n = n^2 x_0^n$ također rješenje gornje rekurzivne relacije. Ukratko, ako je x_0 trostruki korijen karakteristične jednadžbe, onda su $a_n = x_0^n,\ a_n = nx_0^n\ i\ a_n = n^2 x_0^n$ rješenja od (7).

Ako je kompleksni broj x_0 v-struki korijen karakteristične jednadžbe (8), onda svaki od v nizova

$$a_n = x_0^n,\ a_n = nx_0^n, \ldots,\ a_n = n^{v-1}x_0^n,$$

predstavlja rješenje rekurzivne relacije (7).

Teorem 3.2. *Neka su $x_1, x_2, \ldots, x_t$ svi različiti korijeni karakteristične jednadžbe rekurzivne relacije*

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r},\ c_r \neq 0,\ n \geq r,$$

te neka je x_i korijen kratnosti v_i , $i = 1, 2, \ldots, t$. Tada je partikularno rješenje te rekurzije, koje odgovara korijenu x_i, $i = 1, 2, \ldots, t$, dano sa

$$a_n^{(i)} = \lambda_1^{(i)} x_i^n + \lambda_2^{(i)} n x_i^n + \ldots + \lambda_{v_i}^{(i)} n^{v_i - 1} x_i^n = (\lambda_1^{(i)} + \lambda_2^{(i)} n + \ldots + \lambda_{v_i}^{(i)} n^{v_i-1}) x_i^n$$

gdje su $\lambda_j^{(i)}$ neke konstante. Opće rješenje te rekurzije dano je sa $a_n = a_n^{(1)} + a_n^{(2)} + \cdots + a_n^{(t)}$.

Dokaz: Na analogan način kao u raspravi prije Teorema može se pokazati i općenito da, ako je x_i v_i-struki korijen karakteristične jednadžbe, onda su svi $\lambda_j^{(i)} n^{j-1} x_i^n$, $1 \leq i \leq t$,$1 \leq j \leq v_i$, rješenja rekurzije. Budući da su linearne kombinacije rješenja opet rješenja te rekurzije, slijedi da je

$$a_n = a_n^{(1)} + a_n^{(2)} + \ldots + a_n^{(i)} + \ldots + a_n^{(t)}$$

također rješenje.
Neka je sada a_n neko opće rješenje. Tada je a_n potpuno određen početnim uvjetima $a_0 = b_0, a_1 = b_1, \ldots, a_{r-1} = b_{r-1}$. Treba pokazati (kao u dokazu Teorema (3.1)) da možemo odabrati r konstanti $\lambda_j^{(i)}$ (za čvrsto $i \in \{1, 2, \ldots, t\}$ v_i njih, $v_1 + v_2 + \cdots + v_t = r$), tako da

vrijede gornji početni uvjeti, što daje r linearnih jednadžbi sa r nepoznanica. Determinanta tog sustava je jednaka

$$\mathbf{D} = \begin{vmatrix} \overbrace{\begin{matrix} 1 & 0 & \dots & 0 \\ x_1 & x_1 & \dots & x_1 \\ x_1^2 & 2x_1^2 & \dots & 2^{v_1-1}x_1^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{r-1} & \dots & \dots & (r-1)^{v_1-1}x_1^{r-1} \end{matrix}}^{v_1} & \overbrace{\begin{matrix} 1 & 0 & \dots & 0 \\ x_2 & x_2 & \dots & x_2 \\ x_2^2 & 2x_2^2 & \dots & 2^{v_2-1}x_2^2 \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots \end{matrix}}^{v_2} & \begin{matrix} \dots \\ \dots \\ \dots \\ \vdots \\ \dots \end{matrix} & \overbrace{\begin{matrix} 1 & 0 & \dots & 0 \\ x_t & x_t & \dots & x_t \\ x_t^2 & 2x_t^2 & \dots & 2^{v_t-1}x_t^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_t^{r-1} & \dots & \dots & (r-1)^{v_t-1}x_t^{r-1} \end{matrix}}^{v_t} \end{vmatrix}$$

a naziva se **generalizirana Vandermondeova determinanta**, jer se za $v_1 = v_2 = \ldots = v_t = 1$ podudara s Vandermondeovom determinantom. Koristeći se poznatim pravilima za računanje determinanti, dobiva se da je

$$D = \prod_{i=1}^{t} (-x_i)^{\binom{v_i}{2}} \prod_{1 \le i < j \le t} (x_j - x_i)^{v_i v_j}.$$

Kako su brojevi $x_1, x_2, \ldots, x_t$ svi međusobno različiti ne-nul brojevi, $D \neq 0$ pa sustav jednadžbi ima jedinstveno rješenje po $\lambda_j^{(i)}$, čime je Teorem dokazan. ■

Primjer 3.2. *Riješite rekurziju:* $a_n - 7a_{n-1} + 15a_{n-2} - 9a_{n-3} = 0$*, uz početne uvjete* $a_0 = 1, a_1 = 2, a_2 = 3$.

Rješenje: Uvrštavanjem $a_n = x^n$ dobivamo karakterističnu jednadžbu

$$x^3 - 7x^2 + 15x - 9 = 0 \Rightarrow x_1 = 1,\ x_2 = x_3 = 3,$$

sada je opće rješenje te rekurzije

$$a_n = \lambda_1 \cdot 1^n + \lambda_2 \cdot 3^n + \lambda_3 \cdot n \cdot 3^n.$$

Iz početnih uvjeta imamo

$$\begin{aligned} a_0 &= \lambda_1 + \lambda_2 = 1 \\ a_1 &= \lambda_1 + 3\lambda_2 + 3\lambda_3 = 2 \\ a_2 &= \lambda_1 + 9\lambda_2 + 18\lambda_3 = 3 \end{aligned}$$

Jedinstveno rješenje tog sustava je $\lambda_1 = 0,\ \lambda_2 = 1,\ \lambda_3 = \frac{-1}{3}$, pa je traženo rješenje

$$a_n = (1 - \frac{n}{3}) \cdot 3^n$$

Napomena: Ako su x_1 i x_2 par konjugirano-kompleksnih rješenja

$$\begin{aligned} x_1 &= r(cos(\phi) + isin(\phi)) \\ x_2 &= r(cos(\phi) - isin(\phi)) \end{aligned}$$

onda u bazi za opće rješenje umjesto x_1^n i x_2^n možemo uzeti $r^n cos(n\phi)$ i $r^n sin(n\phi)$.

3.2. Linearne nehomogene rekurzije s konstantnim koeficijentima

Linearna nehomogena rekurzivna relacija r-tog reda s konstantnim koeficijentima ima oblik:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r} + f(n), \quad n \geq r \tag{11}$$

gdje su $c_1, c_2, \ldots, c_r$ zadani realni ili kompleksni brojevi i $c_r \neq 0$, a $f_n \colon N \to \mathbb{R}$ $(ili\, \mathbb{C})$ neka zadana funkcija (f je dovoljno zadati na $\{r, r+1, \ldots\}$).

3.2.1. Postupak za rješavanje linearnih nehomogenih rekurzija s konstantnim koeficijentima

Relacije takvog tipa rješavaju se tako da se prvo nađe **opće rješenje pripadne homogene** rekurzije (koja se dobiva izostavljanjem člana $f(n)$, tj. ona za koju je $f(n) = 0$ za svako $n \geq r$), te da se tome doda neko **partikularno rješenje nehomogene** rekurzije. Tako dobiven zbroj je opće rješenje od (11).

Propozicija 3.1. *Svako rješenje rekurzije (11) je zbroj općeg rješenja pripadne homogene rekurzije i partikularnog rješenja nehomogene.*

Dokaz: Neka je a_n^p neko partikularno rješenje od (11). Tada je $a_n - a_n^p$ rješenje pripadne homogene, pa se zbog Teorema (3.2) ono može napisati kao opće rješenje $a_n^{(0)}$ (za neki izbor konstanti λ_{ij}). Dakle, $a_n = a_n^{(0)} + a_n^p$, što smo i trebali pokazati. ∎

Postupak za rješavanje:

1) Odredi se opće rješenje a_n^h pripadne homogene relacije $c_0 a_n + c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r} = 0$ po postupku za homogene relacije, ali ne izračunavamo neodređene koeficijente A_i .

2) Odredi se partikularno rješenje a_n^p nehomogene relacije (prema obliku funkcije $f(n)$, očita je iz tablice).

3) Uvrstimo a_n^p u zadanu rekurziju da odredimo neodređene koeficijente u a_n^p.

4) Opće rješenje nehomogene relacije je $a_n = a_n^h + a_n^p$.

5) Iskoriste se početni uvjeti za određivanje neodređenih koeficijenata koji su preostali u a_n^h.

Primjer 3.3. *Za $r = 1$, (11) je oblika*

$$a_n = c a_{n-1} + f(n). \tag{12}$$

Opće rješenje pripadne homogene rekurzije $a_n = c a_{n-1}$ je $a_n^h = \lambda c^n$. Neka je a_n^p neko partikularno rješenje od (11). Tada je $a_n = a_n^h + a_n^p$ opće rješenje od (11).

$$a_n = \lambda c^n + a_n^p = (c \cdot \lambda c^{n-1}) + (c a_{n-1}^p + f(n)) = c(\lambda c^{n-1} + a_{n-1}^p) + f(n) = c a_{n-1} + f(n)$$

U općem rješenju, konstanta λ odabere se tako da zadovoljava početni uvjet i ne možemo ju pronaći dok nismo odredili a_n^p.
Ako je $c \neq 1$ u (12), onda su poznata rješenja za neke jednostavne funkcije $f(n)$ što je prikazano slijedećom tablicom:

$f(n)$	*partikularno rješenje* a_n^p
d=(*const*)	A
dn	$A_1n + A_0$
dn^2	$A_2n^2 + A_1n + A_0$
d^n	Ad^n
nd^n	$(A_1n + A_0)d^n$

Konstante A, A_0, A_1, A_2 se odrede iz početnih uvjeta. Ako je $f(n)$ suma različitih članova, onda se posebo riješi rekurzija sa svakim članom posebno, ta rješenja se zbroje i to je partikularno rješenje za čitav $f(n)$.

Napomena: Nalaženje partikularnih rješenja je općenito komplicirano, ali u većini slučajeva postoje "recepti". No ipak, postoji slučaj kada prethodno opisana metoda s partikularnim rješenjima nije primjenjiva. To je u slučaju kada je odgovarajuće partikularno rješenje i samo rješenje pripadne homogene rekurzivne relacije. Tada se partikularno rješenje mora tražiti u obliku polinoma stupnja barem n , ili za $f(n) = d^n$ treba pokušati sa And^n itd.

$f(n)$	partikularno rješenje a_n^p
$f(n) = C \cdot d^n$	a) d nije korijen karakteristične jednadžbe $a_n^p = A \cdot d^n$ b) d je korijen karakteristične jednadžbe kratnosti k $a_n^p = A \cdot n^k \cdot d^n$
$f(n)$ polinom stupnja m	a) 1 nije korijen karakteristične jednadžbe $a_n^p = p(n)$-polinom stupnja m s neodređenim koeficijentima $p(n) = \sum_{i=0}^{m} a_i n^i$ b) 1 je korijen karakteristične jednadžbe kratnosti k $a_n^p = n^k \cdot p(n)$, $p(n)$ je polinom stupnja m s neodređenim koeficijentima
$f(n) = C \cdot n^m \cdot d^n$	a) d nije korijen karakteristične jednadžbe $a_n^p = p(n) \cdot d^n$ b) d je korijen karakteristične jednadžbe kratnosti k $a_n^p = n^k \cdot p(n) \cdot d^n$, $p(n)$ je polinom stupnja m s neodređenim koeficijentima

Ovdje C, d i A predstavljaju neke konstante.

4. Sustavi rekurzija i nelinearne rekurzije

Sustavi rekurzivnih relacija mogu se eliminacijom varijabli ili nekim supstitucijama pokušati dovesti do jedne jednadžbe, iz koje onda izračunamo prvi nepoznati niz. Nakon toga računamo ostale nizove.
Linearni homogeni sustavi rekurzivnih relacija često se eliminacijom svode na jednu linearnu homogenu rekurziju. To ćemo pokazati na slijedećem primjeru.

Primjer 4.1. *Nizovi brojeva (a_n) i (b_n) zadovoljavaju sustav rekurzivnih relacija*

$$a_{n+1} = 2a_n - bn$$
$$b_{n+1} = a_n + 4bn.$$

Ako je $a_0 = 2$, $b_0 = 1$, odredite a_n i b_n.

Rješenje: Iz prve rekurzije je $b_n = 2a_n - a_{n+1}$, odnosno $b_{n+1} = 2a_{n+1} - a_{n+2}$. Uvrštavajući dobivene izraze u drugu rekurziju, nakon sređivanja dobivamo da je

$$a_{n+2} = 6a_{n+1} - 9a_n.$$

Opće rješenje te rekurzije je $a_n = \lambda 3^n + \mu n 3^n$, a zbog $a_0 = 2$ i $a_1 = 2a_0 - b_0 = 3$ dobiva se $\lambda = 2$, $\mu = -1$. Dakle, $a_n = (2-n)3^n$. Iz toga se dobiva $b_n = 2a_n - a_{n+1} = (n+1)3^n$.

Primjer 4.2. *Nađite jednu rekurzivnu relaciju kojoj je rješenje niz (a_n) kojem su $a_0 = 0, a_1 = 1, a_2 = 2, a_3 = 1$.*

Rješenje: Tražit ćemo homogenu rekurzivnu relaciju. Prvi problem je odrediti kojeg bi ona trebala biti reda. Rekurzija reda k u sebi ima $k+1$ koeficijent. Da iz razmatranja izbacimo međusobno ekvivalentne rekurzije, podijelimo svaku koeficijentom uz a_{n+k} (tako da sve karakteristične jednadžbe imaju vodeći koeficijent 1). Time smo izjednačili niz rekurzija koje se jedna od druge razlikuju samo u konstantnom faktoru i daju iste nizove kao rješenja. Sada vidimo da su homogene rekurzivne relacije reda k određene s po k koeficijenata.

Budući da imamo zadana četiri člana niza, pokušat ćemo naći rekurziju reda 2 kojoj je neki niz sa zadanim početkom rješenje(razumno je pokušati sa stupnjem rekurzije jednakom polovini broja zadanih članova niza). Ako rekurziju zapišemo kao $a_{n+2} + Aa_{n+1} + Ba_n = 0$, onda uvrštavanjem zadanih članova niza dobivamo sustav jednadžbi:

$$0 = 2 + 1 \cdot A + 0 \cdot B$$

$$0 = 1 + 2 \cdot A + 1 \cdot B,$$

te njeno rješenje $A = -2$, $B = 3$ (očekivali smo $B \neq 0$). Jedna rekurzivna relacija koja zadovoljava zadani početak niza rješenja je $a_{n+2} - 2a_{n+1} + 3a_n = 0$.

Napomena: Da smo pokušali dobiti rekurziju višeg reda, rješenje ne bi bilo jedinstveno. To je i logično, jer je npr. bilo koja linearna kombinacija ove rekurzije i bilo koje rekurzivne relacije koja može zadovoljavati početne uvjete $a_0 = a_1 = a_2 = a_3 = 0$ također rekurzivna

relacija koja zadovoljava uvjet zadatka.
Beskonačno mnogo rješenja pri pokušaju traženja rekurzija reda 2 mogli smo dobiti s nekim drugim početnim uvjetima (npr. 1, 1, 1, 1).
Za neke druge početne uvjete moglo se dogoditi i da dobiveni sustav nema rješenje (npr. 1, 1, 1, 2). Tada bismo morali pokušati s rekurzijom višeg stupnja. Uvijek postoji homogena rekurzivna relacija kojoj je dani konačni niz početni podniz jednog rješenja.

Općenito, **nelinearne rekurzije** su mnogo teže za rješavanje nego linearne. Tehnike za njihovo rješavanje nisu sistematizirane, osim za neke specijalne tipove.
Želimo li riješiti rekurzivnu relaciju oblika

$$a_{n+1} = \frac{pa_n + q}{ra_n + s}$$

uz početni uvjet $a_0 = a$, postupamo ovako:
Neka su (b_n) i (c_n) nizovi koji zadovoljavaju sustav rekurzija

$$b_{n+1} = pb_n + qc_n$$
$$c_{n+1} = rb_n + sc_n,$$

s početnim uvjetima $b_0 = a$ i $c_0 = 1$. Taj sustav rješavamo kao u Primjeru (4.1). Ako je $a_n = \frac{b_n}{c_n}$, onda je $a_0 = \frac{b_0}{c_0} = a$ i imamo

$$a_{n+1} = \frac{b_{n+1}}{c_{n+1}} = \frac{pb_n + qc_n}{rb_n + sc_n} = \frac{pb_n/c_n + q}{rb_n/c_n + s} = \frac{pa_n + q}{ra_n + s},$$

pa je (a_n) traženi niz.

Neke nelinearne rekurzije svode se na linearne pogodnom supstitucijom ili pogodnom supstitucijom nakon nekih transformacija. To nam pokazuju idući primjeri.

Primjer 4.3. *Riješite rekurzivnu relaciju $a_{n+1} = a_n(2 - ca_n)$, uz početni uvjet $a_0 = a$.*

Rješenje: Za $c = 0$ relacija je oblika $a_{n+1} = 2a_n$, slijedi da je $a_n = a2^n$. Ako je $c \neq 0$, onda se naša relacija može transformirati na sljedeći način :

$$ca_{n+1} = ca_n(2 - ca_n) \Longleftrightarrow ca_{n+1} = (1 - (1 - ca_n))(1 + (1 - ca_n)) = 1^2 - (1 - ca_n)^2,$$

tj. $1 - ca_{n+1} = (1 - ca_n)^2$. Supstitucijom $1 - ca_n = b_n$ dobivamo $b_{n+1} = b_n^2$, $b_0 = 1 - ac$. Odavde slijedi (direktno ili supstitucijom $c_n = \log_2 b_n$) da je $b_n = (1 - ac)^{2^n}$, pa je $a_n = \frac{1}{c}(1 - (1 - ac)^{2^n})$

Primjer 4.4. *Riješite rekurzivnu relaciju $a_{n+1} = \frac{1}{2}\left(a_n + \frac{c}{a_n}\right)$, $c > 0$, uz početni uvjet $a_0 = a$.*

Rješenje: Transformirajmo našu relaciju na sljedeći način :

$$\frac{a_{n+1} - \sqrt{c}}{a_{n+1} + \sqrt{c}} = \frac{\frac{1}{2}(a_n + \frac{c}{a_n}) - \sqrt{c}}{\frac{1}{2}(a_n + \frac{c}{a_n}) + \sqrt{c}} = \frac{a_n^2 - 2a_n\sqrt{c} + c}{a_n^2 + 2a_n\sqrt{c} + c} = \left(\frac{a_n - \sqrt{c}}{a_n + \sqrt{c}}\right)^2.$$

Uvedimo supstituciju $\frac{a_n-\sqrt{c}}{a_n+\sqrt{c}} = b_n$. Tada dobivamo $b_{n+1} = b_n^2$, $b_0 = \frac{a-\sqrt{c}}{a+\sqrt{c}}$. Odatle (iteriranjem ili supstitucijom $c_n = \log_2 b_n$) lako dobivamo $b_n = b_0^{2^n}$, što konačno daje

$$a_n = \frac{\sqrt{c}(1+b_0^{2^n})}{(1-b_0^{2^n})}.$$

Napomena: Ova rekurzija je tzv. **Heronov algoritam** za sukcesivne aproksimacije broja $\sqrt{c}$. Prelaskom na limes u rekurziji dobivamo

$$\lim_{n\to\infty} a_n = \sqrt{c}$$

Primjer 4.5. *Riješite slijedeće rekurzivne relacije svodeći ih pogodnom supstitucijom na linearne:*
(a) $a_{n+1}^2 - 2a_n^2 = 1,\ a_0 = 2,$
(b) $a_{n+1} - na_n = n!,\ a_0 = 2,$
(c) $(n+1)a_{n+1} + na_n = 2^n,\ a_0 = 59.$

Rješenje:

(a) U ovoj rekurzivnoj relaciji pojavljuju se samo kvadrati članova niza (a_n), pa možemo označiti $b_n := a_n^2$. Relacija prelazi u $b_{n+1} - 2b_n = 1$ uz početni uvjet $b_0 = 2^2 = 4$. Opće rješenje te rekurzivne relacije je $b_n = A \cdot 2^n - 1$, a uvrštavanjem početnog uvjeta dobiva se rješenje nove rekurzije: $b_n = 5 \cdot 2^n - 1$. Rješenje zadatka je $a_n = \sqrt{b_n} = \sqrt{5 \cdot 2^n - 1}$.

(b) Do rekurzije s konstantnim koeficijentima dolazimo supstitucijom $b_n = a_n/(n-1)!$ za $n \geq 1$. Nova rekurzija je tada $b_{n+1} - b_n = 1$, a početni uvjet $b_1 = a_1 = 1$. Rješenja su $b_n = n$, odnosno $a_n = n!$ (za sve $n \geq 0$).

(c) Svaki član niza pojavljuje se pomnožen sa svojim indeksom, pa možemo staviti $b_n := na_n$. Dobivamo rekurziju $b_{n+1} + b_n = 2^n$ i početni uvjet $b_0 = 0$. Rješenje te rekurzije je $b_n = ((-1)^n + 2^n)/3$, a rješenje polazne rekurzije $a_n = ((-1)^n + 2^n)/(3n)$ za $n > 0$ i $a_0 = 59$.

Razne rekurzivne relacije se vrlo lako mogu rješavati tehnikom funkcija izvodnica[7]. Ideja se sastoji u tome da iz rekurzivne relacije dobijemo funkciju izvodnicu koja zamjenjuje čitav niz zadan rekurzivno. Trik koji se primjenjuje pri rješavanju rekurzivnih relacija je jednostavan. Relacija se pomnoži s x^n ili nekom višom pogodnom potencijom, a zatim se tako dobiveni izrazi za $n = 0, 1, 2, \ldots$ zbroje. Zbrajanjem se dobiju tražene funkcije izvodnice. Nakon toga po potrebi možemo izračunati rješenja - koeficijente u funkciji izvodnici. Često iz funkcija izvodnica možemo naći i novu rekurziju različitu od one od koje smo pošli.

[7]Neka je $a_0, a_1, \ldots, a_n, \ldots$ niz realnih ili kompleksnih brojeva. Pripadna funkcija izvodnica (FI) je formalni red potencija

$$a(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n + \cdots = \sum_{n\geq 0} a_n t^n$$

5. Rekurzivne relacije za neke kombinatorne brojeve

Neka je $f\colon N \to C$ zadana funkcija. Tada se rekurzije $a_n = f(n)a_{n-1}$ rješavaju tako da se sve sljedeće relacije pomnože:

$$\begin{aligned} a_n &= f(n)a_{n-1} \\ a_{n-1} &= f(n-1)a_{n-2} \\ a_{n-2} &= f(n-2)a_{n-3} \\ &\vdots \\ a_2 &= f(2)a_1 \\ a_1 &= f(1)a_0 \end{aligned}$$

To se može shvatiti kao "teleskopiranje". Dobiva se

$$a_n = f(1)f(2)\ldots f(n)a_0.$$

U vezi s tim jednostavnim tipom rekurzija spomenit ćemo dva poznata kombinatorna problema koji će kao rješenja dati tzv. **Catalanove brojeve**[8].

5.1. Catalanovi brojevi

n-ti **Catalanov broj** jednak je

$$C_n = \frac{1}{n+1}\binom{2n}{n},$$

a prvih nekoliko su (uz konvenciju $C_0 = 1$):

n	0	1	2	3	4	5	6	7	8	...
C_n	1	1	2	5	14	42	132	429	1430	...

Ove zanimljive brojeve razmatrali su Euler[9] i Segner [10], gotovo čitavo stoljeće prije Catalana. Proučavajući problem triangulacije konveksnog n-terokuta, Segner je postavio rekurzivnu relaciju, a Euler prvi uspješno riješio i 1760. godine došao do općeg izraza za broj triangulacija. Zanimljivo je napomenuti kako je Euler problemu pristupio alatima koji se pripisuju modernoj kombinatorici, izračunavši tako funkciju izvodnicu za broj triangulacija i zatvorenu formulu iz nje. Ipak, u čast Catalanu, koji je izveo i dokazao mnoga svojstva i identitete ovih brojeva, oni se danas zovu njegovim imenom. Catalanovi brojevi javljaju se u mnoštvu naizgled nepovezanih kombinatornih problema.

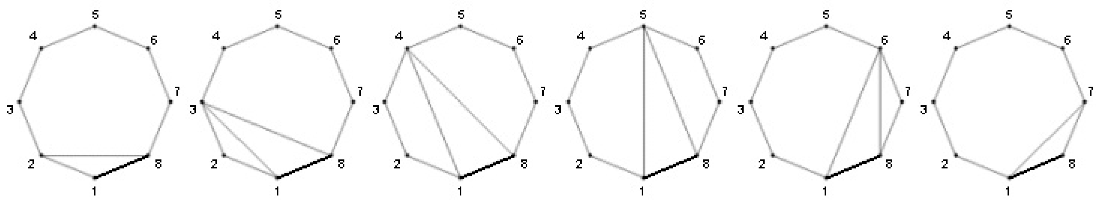
5.1.1. Problem triangulacije konveksnog n-terokuta

Razmatra se broj načina (označimo taj broj s T_n) na koji je moguća maksimalna dekompozicija konveksnog n-terokuta na $n-2$ trokuta (otuda ime triangulacija). Da bismo ga

[8]Eugene Charles Catalan (1814.-1894.), francuski matematičar
[9]Leonhard Euler (1703.-1783.), švicarski matematičar
[10]Johann Andreas von Segner (1704.-1777.)

triangulirali, potrebno je povući $n-3$ dijagonala koje se ne smiju sjeći. Jedna dijagonala dijeli ga na dva dijela, dvije na tri i tako dalje indukcijom po n.

Razmotrimo problem induktivno i počnimo s trokutom. $T_3 = 1$ jer je trokut već (jednoznačno) trianguliran, a dodatno još definiramo $T_2 := 1$. Za konveksan četverokut ($n = 4$) moramo povući jednu dijagonalu. To se može učiniti na dva načina (jer takav četverokut ima dvije dijagonale), dakle $T_4 = 2$. Za peterokut ($n = 5$) rješenje je manje očigledno, postoji 5 načina triangulacije. Nađimo sad opće rješenje za broj triangulacija n-terokuta T_n.
Svaka stranica n-terokuta je dio točno jednog trokuta triangulacije. Za prebrojavanje se koristi rekurzija i slijedeći algoritam. Nasumce odabiremo i fiksiramo jednu od stranica te brojimo triangulacije u kojima sudjeluje svaki od trokuta podignutih nad tom stranicom. Za k-tu točku kao vrh trokuta, zdesna je ostao $(n-k+1)$-terokut, koji možemo triangulirati na T_{n-k+1} načina, a s lijeva k-terokut koji možemo triangulirati na T_k načina (vidi sliku 5.2).
Budući da su izbori triangulacije izdvojenih mnogokuta međusobno neovisni, vrijedi kom-



Slika 5.2: Fiksiramo jednu stranicu i nad njom konstruiramo trokute (slika preuzeta sa [10])

binatorni princip produkta te je za odabranu točku k taj broj jednak T_kT_{n-k+1}. Izbor te točke također možemo učiniti na više (neovisnih) načina, preostaje nam samo prosumirati po svim mogućim vrijednostima k. Konačan oblik za T_n glasi:

$$T_n = \sum_{k=2}^{n-1} T_kT_{n-k+1}$$

5.1.2. Problem zagrada

Postavlja se pitanje na koliko načina se može u nizu rasporediti $2n$ zagrada tako da, brojeći s lijeva, ni u jednom trenutku ne nabrojimo (strogo) više zatvorenih nego otvorenih zagrada. Npr. zagrađivanje ()() je valjano, a ())(nije jer na poziciji 3 ima veći broj zatvorenih nego otvorenih zagrada. Prva zagrada mora biti otvorena, slijedeća može biti otvorena ili zatvorena, iduća pak zatvorena samo ako prethodna nije bila, itd. Problem je naći ukupan broj zagrađivanja, tj. treba izračunati na koliko se načina može pomnožiti n brojeva, ali da su u nekom čvrstom poredku $x_1x_2\ldots x_n$. Traženi broj je Z_n, npr. za $n = 3$ imamo 4 broja i pet načina zagrađivanja (zadnje množenje ne zagrađujemo jer je ono jednoznačno određeno)

$$(x_1x_2)(x_3x_4) = ((x_1x_2)x_3)x_4 = (x_1(x_2x_3))x_4 = x_1((x_2x_3)x_4) = x_1(x_2(x_3x_4))$$

Problem je također ekvivalentan broju mogućih valjanih ugniježđenja n blokova u nekim programskim jezicima (npr. u programskom jeziku C). Točnije, prebrojavamo valjane rasporede n parova zagrada koji omeđuju blokove. Odaberemo li prvi blok duljine k, drugi mora biti duljine $n-k$. Stoga prvi možemo zagraditi na Z_k, a drugi blok na Z_{n-k} načina i to nezavisno jedan od drugoga. Prema pravilu produkta, ako je duljina prvog bloka k, ima $Z_k Z_{n-k}$ načina zagrađivanja. Sumira se po svim mogućim vrijednostima k. Blokovi moraju biti neprazni i zato k ne poprima vrijednosti veće od 1 ni one manje od $n-1$. Konačno, Z_n glasi:

$$Z_n = \sum_{k=1}^{n-1} Z_k Z_{n-k}$$

5.2. Bellov i Stirlingov broj

Definicija 5.1. *n-**ti Bellov broj** P_n definira se kao broj particija n-članog skupa* $(n \in \mathbb{N})$ *i* $P_0 := 1$.

Teorem 5.1. *Bellovi brojevi zadovoljavaju za sve* $n \geq 0$ *rekurziju*

$$P_{n+1} = \binom{n}{0} P_0 + \binom{n}{1} P_1 + \binom{n}{2} P_2 + \cdots + \binom{n}{n} P_n = \sum_{j=0}^{n} \binom{n}{j} P_j$$

Dokaz: Neka je S skup sa $|S| = n+1$ elemenata na kojem se obavljaju sve moguće particije (u disjunktne, neprazne podskupove, tj.blokove) i neka je $x \in \mathbb{S}$. Blok koji sadrži x neka ima osim x još k elemenata. On se može izabrati na

$$\binom{n+1-1}{k+1-1} = \binom{n}{k}$$

načina. Na preostalih $(n+1)-(k+1) = n-k$ elemenata imamo P_{n-k} particija. Prema tome, broj particija od S kod kojih blok koji sadrži x ima $k+1$ elemenata jednak je, prema principu produkta, $\binom{n}{k} P_{n-k}$. To je istina i za $k = n$ jer je $P_0 = 1$. Stoga je prema principu sume

$$P_{n+1} = \sum_{k=0}^{n} \binom{n}{k} P_{n-k} = \sum_{k=0}^{n} \binom{n}{n-k} P_{n-k} = \sum_{j=0}^{n} \binom{n}{j} P_j$$

■

Ta rekurzivna relacija je tzv. **binomnog tipa** jer sadrži binomne koeficijente.

Definicija 5.2. *Rastav skupa* $1, 2, \ldots, n$ *na* k *nepraznih disjunktnih podskupova zovemo particija skupa. Broj particija skupa* $1, \ldots, n$ *na* k *dijelova označava se sa* $S(n,k)$ *i zove Stirlingov broj druge vrste. Dijelove (podskupove) particije zovemo blokovima.*

Napomena: Iz definicije slijedi $S(n,k) = 0$ za $1 \leq n < k$. Dogovoreno je $S(0,0) = 1$.

Očito se Bellovi brojevi dobiju kao sume Stirlingovih brojeva:

$$P_n = \sum_{k=0}^{n} S(n,k)$$

Teorem 5.2. *Stirlingovi brojevi druge vrste $S(n,k)$ zadovoljavaju sljedeće rekurzije:*

(a) $S(n,k) = S(n-1,k-1) + kS(n-1,k),\ \ n,k \in \mathbb{N};\ \ S(n,0) = S(0,k) = 0,\ \ S(0,0) = 1$

(b) $S(n,k) = \displaystyle\sum_{m=0}^{n-1} \binom{n-1}{m} S(m,k-1)$

(c) $S(n,k) = \displaystyle\sum_{m=k}^{n} S(m-1,k-1)k^{n-m}$

Dokaz: Ove rekurzije dokazujemo kombinatorno.
(a) Neka je $|X| = n$ i $x \in X$. Uočimo dvije vrste k-particija od X: one u kojima se javlja $\{x\}$ kao blok i one u kojima se ne javlja. Broj k-particija od X u kojima se javlja $\{x\}$ kao blok je $S(n-1,k-1)$, jer je to broj načina da se $(n-1)$-člani skup $X\backslash\{x\}$ particionira u $k-1$ blokova. Ako x pripada nekom bloku zajedno s još nekim drugim elementima, odstranimo ga za trenutak. Tako dobivamo particiju $(n-1)$-članog skupa $X\backslash\{x\}$ u k blokova, a takvih ima $S(n-1,k)$. Budući da x može pripadati bilo kojem od k blokova takve particije, ukupan je broj načina da se x ponovo umetne pa da dobijemo k-particiju od X jednak $kS(n-1,k)$. Zbrajanjem tih dviju mogućnosti dobivamo $S(n,k) = S(n-1,k-1) + kS(n-1,k), n,k \in \mathbb{N}; S(n,0) = S(0,k) = 0, S(0,0) = 1.$

(b) Neka je $|X| = n$ i $x \in \mathbb{X}$. Odstranimo za trenutak blok B particije od X koji sadrži x. Neka on ima r elemenata. Tim uklanjanjem preostaje $(n-r)$-člani skup $X\backslash B$, koji je s preostalim blokovima particioniran u $k-1$ blokova. Kako imamo ukupno $\binom{n-1}{r-1}$ načina da odaberemo takav blok B, a broj particija od $X\backslash B$ na $k-1$ blokova jednak je $S(n-r,k-1)$.Dobivamo (zbog principa sume) da je

$$S(n,k) = \sum_{r=1}^{n} \binom{n-1}{r-1} S(n-r,k-1) = \sum_{m=0}^{n-1} \binom{n-1}{m} S(m,k-1).$$

■

Relacije (a) i (b) omogućuju nam lakše izračunavanje brojeva $S(n,k)$ za male vrijednosti n i k.

5.3. Dvoindeksne rekurzije

To su rekurzije za nizove $(a_{n,k})_{n,k\geq 0}$ koji ovise o dvama cijelim parametrima $n,k \geq 0$. Spomenimo samo neke takve jednostavne rekurzije i njihova rješenja, npr. rekurzije

$$a_{n,k} = a_{n-1,k-1} + a_{n-1,k}, \qquad a_{n+1,k+1} = \sum_{l=k}^{m} a_{l,k}$$

su zadovoljene za $a_{n,k} = \binom{n}{k}$.
Stirlingovi brojevi druge vrste zadovoljavaju rekurziju $s_{n,k} = s_{n-1,k-1} + ks_{n-1,k}$, a Stirlingovi brojevi prve vrste zadovoljavaju rekurziju $c_{n,k} = c_{n-1,k-1} + (n-1)c_{n-1,k}$, dok Eulerovi brojevi zadovoljavaju $e_{n,k} = (k+1)e_{n-1,k} + (n-k)e_{n-1,k-1}$.

6. Rekurzije u programiranju

Rekurzija je važan koncept u računalnoj znanosti jer se mnogi algoritmi mogu pomoću nje najbolje prikazati. Rekurzija nastaje kada funkcija (procedura) poziva samu sebe direktno ili indirektno. Indirektna rekurzija nastaje kada jedna funkcija poziva drugu funkciju, iz koje se ponovo poziva originalna funkcija. Ukoliko se problem može riješiti bez rekurzije to je bolji odabir jer s rekurzijom je vrijeme izvođenja obično dulje, iako su rekurzivni programi kraći. Rekurzija je dodatno neučinkovita i zbog višestrukog računanja nekih vrijednosti (ponavljanja). Neki jezici (npr. stare verzije jezika *Fortran*) ne podržavaju rekurziju.
Pojednostavljeno rečeno, rekurzija je algoritam koji poziva sam sebe, pri čemu do rješenja problema dolazimo rješavanjem podproblema.
Rekurzivne funkcije treba pisati tako da je svakim novim pozivom posao koji obavlja funkcija sve bliži kraju, tj. konvergira konačnom rješenju. Rekurzija nema petlji. Ponavljanje se postiže tako da funkcija poziva samu sebe sve dok se ne dođe do osnovnog slučaja. Koristi se struktura podataka (stog) za pohranjivanje rezultata i povratak iz rekurzije.

Karakteristike rekurzije:
Osnovni slučajevi: Uvijek moraju postojati osnovni slučajevi koji se rješavaju bez rekurzije.
Korak rekurzije: Za slučajeve koji se rješavaju rekurzivno, svaki sljedeći rekurzivni poziv mora biti približavanje osnovnim slučajevima.
Pravilo projektiranja: Podrazumijeva se da svaki rekurzivni poziv funkcionira.
Pravilo neponavljanja: Ne smije se dopustiti da se isti problem rješava odvojenim rekurzivnim pozivima (to rezultira nagomilavanjem posla).

Najjednostavniji primjer razlike standarnog pristupa i rekurzivnog pristupa je funkcija koja računa faktorijel danog broja. Primjeri rekurzivnih funkcija (1) jednog nezavisnog i (2) jednog koji izgleda baš kao i početni posao.
Tada se nezavisni dio izračuna (očito nerekurzivno, bar što se tiče početne funkcije), a za ostatak posla pozove se ponovno sama funkcija. Ako grupiramo članove na ovaj način: $n! = n \cdot [(n-1) \cdot (n-2) \cdot .... \cdot 2 \cdot 1]$, tada je $n! = n \cdot (n-1)!$
Za $n!$ treba izračunati $(n-1)!$, što je posao (2), te to pomnožiti s n što je posao (1). Kada se ovdje prestaje sa rekurzivnim pozivima? Treba izračunati funkciju za par brojeva.
3! = 3 * 2! ... rekurzivni poziv
2! = 2 * 1! ... rekurzivni poziv
1! = 1 ... nema rekurzije!
Iz priloženog primjera vidi se da rekurzivni pozivi prestaju onog trena kada argument funkcije postane jednak 1, tj. matematika nam govori i da je 0! isto tako jednako 1, pa možemo reći da rekurzivni pozivi prestaju za 1 ili za 0. No, što je sa negativnim brojevima? Koliko je $(-1)!$? Relacija kojom smo opisali način izračuna faktorijela za negativne brojeve divergira, jer ako od -1 počnemo oduzimati jedinice, nikada nećemo stići do nekog pozitivnog broja ili nule, te će i naša rekurzivna funkcija, kada je jednom napišemo, biti divergentna za takve

brojeve. Zato je potrebno proširiti gornju definiciju faktorijela sa uvjetom koji matematički možda i nije točan, ali će osigurati da poziv funkcije ne zamrzne računalo ako se pozove sa negativnim brojem. Recimo da ako je argument funkcije manji od 0, funkcija je jednaka 0. Faktorijele se definiraju za cijele brojeve i jako brzo rastu.

Jedan od jednostavnih rekurzivnih algoritama je izračunavanje $n!$
Napišimo funkciju $f(n) = n!$ rekurzivno

$$f(n) := \begin{cases} 0 & \text{, ukoliko je } n < 0 \\ 1 & \text{, ukoliko je } n = 0 \\ n \cdot (n-1)! & \text{, ukoliko je } n > 0 \end{cases}$$

$f(n) = n \cdot f(n-1)$
za n=0: $0! = 1$ (Temeljno rekurzije)
$1! = 1$ Osnovni slučaj
za $n > 0$: $n! = n * (n-1)!$ (Rekurzivno pravilo)
Rekurzivna definicija: Sve se vrijednosti mogu izračunati pomoću gornjeg rekurzivnog pravila: $1! = 1 * 0! = 1$, $2! = 2 * 1! = 2$, $3! = 3 * 2! = 6$ ... itd. $n! = n * (n-1)!$
Računalo računa: fact (4)=4 *fact (3)=4*3 *fact (2)=4*3 *2 *fakt (1)=4*3 *2 *1 *fakt (0)

Algoritam 6.1. *Kod za Faktorijel funkciju*

```
int fact(int n)
{
if (n < 0) /*Ako je broj negativan, vrati nulu*/
return 0;
else if (n == 0)
return 1;  /*Ako je broj jednak 0 ili 1, vrati 1*/
else
return n*fact(n-1); /*Inače koristi rekurzivnu relaciju*/
}
```

Primjer 6.1. *Program računa n-ti broj Fibonaccijevog niza. Rješenje: Broj zečeva se povećava prema ovom nizu 0,1,1,2,3,5,8,13,21,... (koji je sljedeći?)*
Definicija funkcije(rekurzivno):

$$fib(n) := \begin{cases} 0 & , \textit{za } n = 0 \\ 1 & , \textit{za } n = 1 \\ fib(n-1) + fib(n-2) & , \textit{za } n \geq 2 \end{cases}$$

Algoritam 6.2. *Kod za Fibonacci funkciju*

```
int fib(int n)
{
if ( n == 0 )
return 0;
else if ( n == 1 )
```

```
return 1;
else
return fib(n-1)+fib(n-2);
}
```

Jedan od najstarijih algoritama je **Euklidov postupak** za pronalaženje najveće zajedničke mjere (djelitelja) (n.z.m) dva prirodna broja. Treba odrediti, npr. mjeru prirodnih brojeva a i b. Računalo ne zna odrediti mjeru brojeva a i b, ali zna mjeru nekog drugog sličnog para, npr. $(a-1,b),(a,b-1),(a/2,b),\ldots$ Trebamo naći vezu između mjere neka dva slična para prirodnih brojeva i brojeva a i b. Ako su brojevi jednaki, onda je njihova mjera jedan od brojeva (a ili b), a ako su brojevi različiti onda je njihova mjera jednaka mjeri sljedećeg para brojeva.

Primjerice, nzm(22,8) = nzm(8,6) = nzm(6,2) = nzm(2,0) = 2,
nzm(21,13) = nzm(13,8) = nzm(8,5) = nzm(5,3) =nzm(3,2) = nzm(2,1) = nzm(1,0) = 1,
nzm (0,21) = nzm (21,0) = 21.

Uvjet prekida: a = b, mjera je npr. a
Rekurzivna relacija: mjera (a, b) = mjera (min (a, b), $|a-b|$) funkcija mjera (a, b)
Algoritam rješenja:
ako je a = b onda
mjera := a
inače ako je $a > b$ onda
mjera := mjera (a – b, b)
inače mjera := mjera (b – a, a)

Algoritam 6.3. *Kod za Euklida*

```
int nzm (int a, int b)
{
if (b == 0) return a;
return nzm (b, a % b); /*nzm = najveća zajednička mjera od b i ostatka dijeljenja a sa b*/
}
```

Primjer za **pogrešku**:

Algoritam 6.4. *int los (int n)*

```
{
if (n == 0) return 0;
return los (n / 3 + 1) + n - 1;
}
```

Pogreška je u tome što za vrijednost $n = 1$ rekurzivni poziv je opet s argumentom 1. Nema napredovanja prema osnovnom slučaju. Program ne radi ni za druge vrijednosti argumenta. Npr. za $n = 4$, rekurzivno se poziva *los* s argumentom 4/3+1=2, zatim 2/3+1=1 i dalje stalno 1/3+1=1

6.1. Podijeli pa vladaj (Devide and Conquer)

Jedna posebna klasa rekurzivnih relacija koje se često javljaju u analizi nekih (rekurzivnih) algoritama zahtijeva pažnju. Takvi algoritmi tipa "podijeli pa vladaj" rekurzivno razbijaju problem na dva ili više podproblema istog tipa, sve dok ovi ne postanu tako jednostavni da se mogu riješiti direktno. Rješenja ovih podproblema se onda kombiniraju kako bi dali rješenja sveukupnog problema. Ova je tehnika baza za mnoge djelotvorne algoritme svih vrsta, kao što je sortiranje i diskretne Fourierove transformacije.

Za složenost algoritma dobivamo rekurzivnu relaciju izraženu u veličini zadaće. Ta relacija može biti jednadžba ako radimo preciznu analizu složenosti. No, često nas zanima samo red veličine složenosti, pa pojednostavljujemo neke elemente složenosti, što vodi na nejednadžbe. Obično trebamo gornju ogradu, pazeći da ona ne bude pregruba. Katkad, istovremeno radimo i ocjenu odozdo, s ciljem da preciznije opišemo ponašanje složenosti. Funkcije $f(n)$ su gotovo uvijek relativno jednostavnog oblika, jer želimo kratku informaciju o složenosti, pa nevažne detalje odmah zanemarujemo.

Ukupan broj koraka a_n potrebnih da se takvim algoritmom procesira n ulaznih podataka (inputa), često zadovoljava rekurzivnu relaciju tipa $a_n = ca_{n/d} + f(n)$ (d je djelitelj od n). Sljedeća tablica daje oblik rješenja te rekurzije za neke vrijednosti c, d i $f(n)$.

c, d	$f(n)$	a_n
$c = 1$	a	$\lceil log_d n \rceil$
$c = d$	a	$A_n + B$
$c < d$	an	An
$c = d$	an	$\sim n \log_d n$
$c > d$	an	$\sim n^{\log_d c}$

Pritom $\sim$ shvaćamo ovako: a_n je proporcionalan s navedenim izrazom za velike n. D su to zaista rješenja gornje rekurzije, može se lako pokazeti indukcijom.

Primjer 6.2. *Nađite i riješite rekurziju za broj a_n usporedbi od po dvaju brojeva, potreban da se u skupu S od n različitih realnih brojeva nađu najmanji i najveći broj ako je n potencija od 2.*

Rješenje: Ako je $|S| = 2$, onda je očito $a_2 = 1$. Neka su m_1 i M_1 najmanji i najveći brojevi redom u prvoj polovici od S (prvih $n/2$ brojeva), a m_2 i M_2 najmanji i najveći brojevi u drugoj polovici skupa S. Tada ove dvije usporedbe: m_1 i m_2, te M_1 i M_2 određuju najmanji i najveći broj u čitavom S. Stoga je $a_n = 2a_{n/2} + 2$. Iz prethodne tablice dobivamo da je rješenje oblika $a_n = A_1 n + A_2$, pa uvrštavanjem u tu rekurziju i iz početnog uvjeta $a_2 = 1$ dobivamo da je $a_n = \frac{3}{2}n - 2$. Ta se metoda katkad zove i **binarno pretraživanje**.

6.1.1. Merge Sort (sortiranje sažimanjem)

Algoritam konstruiran metodom "podijeli, pa vladaj". Zadano polje podijeli se na dva manja polja podjednakih duljina. Ta dva manja polja zasebno se sortiraju rekurzivnim pozivima istog algoritma. Zatim se mala sortirana polja sažimaju u jedno (sortirano). MergeSort uzima kao ulaz jednu listu i vraća listu sortiranih elemenata te liste. Radi tako što od originalne liste napravi dvije liste, otprilike podjednake dužine, te ponovo sebe pozove na te obje liste. Rekurzija se zaustavlja kada dobijemo listu dužine 1. Nakon što sve svedemo na skup lista dužine jedan, spajamo ih ponovo, na pametan način, tj. sortirano.

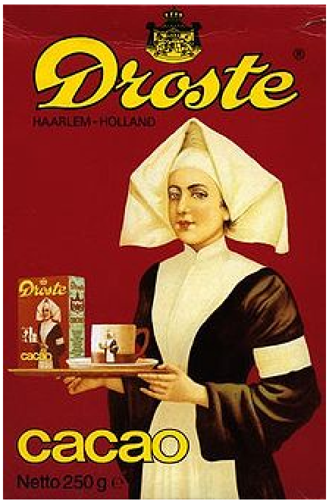
6.1.2. Quick Sort (brzo sortiranje)

Spada također u klasu algoritama "podijeli pa vladaj" i rekurzivan je. U teoriji je jednostavan, no složen u implementaciji. Osnovna ideja je slijedeća: potrebno je rekurzivno primjenjivati slijedeći postupak: polje treba podijeliti na dva podpolja, na taj način da odaberemo jedan stožerni element, te polje uredimo tako da sve elemente koji su manji od stožernog stavimo ispred stožernog, a sve elemente koji su veći od stožernog stavimo iza stožernog. Na ovako nastala dva podpolja (od prvog elementa do stožernog, ali bez stožera, i od prvog elementa iza stožera do kraja) potrebno je ponovno primijeniti Quick Sort rekurzivno.

7. Rekurzija u slikama

- ***Droste* efekt**

  Termin *Droste* efekta dolazi od pjesnika i kolumnista Nico Scheepmaker, krajem 70-tih godina. Naziv Droste dobio je kasnije, po njemačkom brendu tople čokolade, na čijem je pakovanju bila slika sestre koja na pladnju poslužuje šalicu i pakovanje istog brenda tople čokolade. Na tom drugom pakovanju tople čokolade je opet jednaka slika sestre i to se ponavlja unedogled (v.sl. 7.3).



Slika 7.3: Vizualni oblik rekurzije (slika preuzeta sa[4])

- **Gotovo paralelna ogledala**

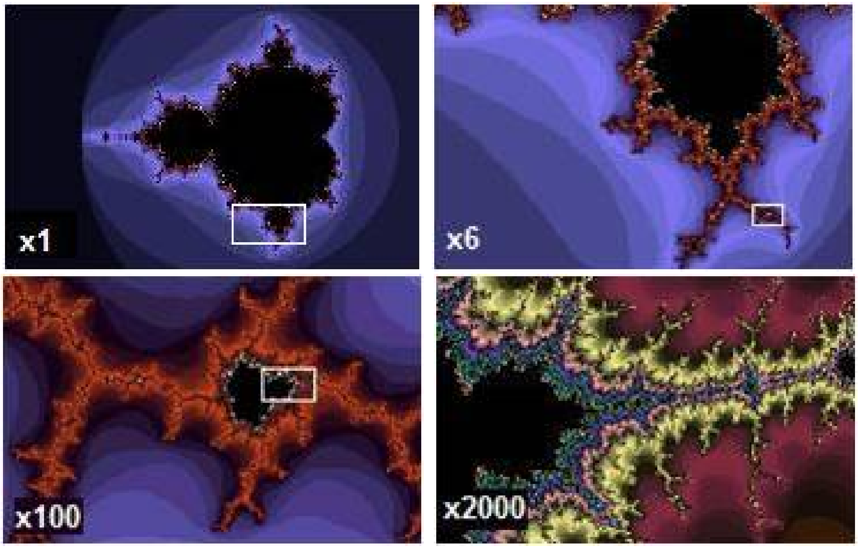
  Naziv rekurzija se općenitije koristi za opis procesa ponavljanja objekata na samosličan način. Primjerice, kada su površine dvaju zrcala gotovo uzajamno paralelne, ugniježđene slike koje se pojavljuju su jedan oblik rekurzije.

- **Rekurzivni fraktali**

  Fraktali su skupovi točaka kojima je fraktalna dimenzija veća od topološke dimenzije. Konačne su površine i beskonačnog opsega. Naziv im je dao Benoit Mandelbrot 1975. godine od latinskog pridjeva *fractus* što znači razlomljen. Fraktali imaju tri važna svojstva: sličnost samome sebi, fraktalnu dimenziju i oblikovanje iteracijom. Fraktalnu dimenziju sebi sličnog skupa definiramo s: $d = log(P)/log(s)$ gdje se objekt (skup) sastoji od P kopija samog sebe za faktor s. Ova definicija vrijedi samo za sebi slične skupove. Skup S ima topološku dimenziju d, ako svaka točka u S ima po volji malu okolinu čije granice dodiruju S u skupu dimenzije $d - 1$, a d je najveći pozitivni cijeli broj za kojeg ovo vrijedi. Fraktale je moguće klasificirati prema načinu nastajanja na iterativne, rekurzivne i slučajne (random).

  - Iterativni fraktali (Kochova krivulja) posjeduju najveći stupanj samosličnosti tzv. potpunu samosličnost. Bez obzira na to koji dio smo uvećali uvijek ćemo dobiti sliku koja je identična početnoj.

- Rekurzivni fraktali (Mandelbrotov skup, (v.sl. 7.4)) To su fraktali koje dobivamo iz rekurzivnih relacija, funkcija koja pozivaju same sebe. Oni posjeduju svojstvo kvazisamosličnosti, što znači da je fraktal približno, ali ne potpuno jednak na različitim razinama.
- Slučajni (random) fraktali (cvjetača) posjeduju najmanji stupanj samosličnosti, tzv. statističku samosličnost. Nalazimo ih svugdje u prirodi.



Slika 7.4: Fraktal je objekt koji pokazuje sličnost samome sebi - uvećanjem jednog dijela fraktala dobivamo strukturu koja je umanjena verzija početnog dijela fraktala (slike preuzete sa [8])

Ovo poglavlje završit ću s jednom rekurzivnom šalom koju sam pronašla u [4].
Općenita šala je slijedeća “definicija” rekurzije.

Rekurzija

Vidi “Rekurzija”

Varijacija na ovu šalu je:

Rekurzija

Ako i dalje ne shvaćaš, pogledaj “Rekurzija”, koja se zapravo izvršava čim čitatelj “shvati”.

Literatura

[1] D. VELJAN, *Kombinatorna i diskretna matematika*, Algoritam, Zagreb 2001.

[2] D. VELJAN, *Kombinatorika s teorijom grafova*, Školska knjiga, Zagreb 1989.

[3] M. CVITKOVIĆ, *Kombinatorika*, Element, Zagreb 1994.

[4] http://en.wikipedia.org/wiki/Recursion

[5] http://web.math.hr/nastava/kidm/rekurzije.pdf

[6] http://www.fesb.hr/ borka/files/DM-7p-06.pdf

[7] http://zvrba.net/writings/dis.pdf

[8] http://en.wikipedia.org/wiki/Fractal

[9] http://www.blog.republicofmath.com/archives/3308

[10] http://e.math.hr/old/catalan/index.html