Рекурентне релације и алгоритамска парадигма *подели па владај*

Константин Ранисављевић
Број индекса: 186/2025
Математички факултет
Универзитет у Београду

24. фебруар 2026.

Садржај

Увод

Многе активности без којих данашњу свакодневицу не бисмо могли ни замислити извршавамо брже него икада пре. То нам данас омогућавају рачунари и њима прилагођени алгоритми који су некада представљали доста спорије процедуре. Ипак, примећујемо да, како сложеност ових алгоритама и количина података који у њима учествују расту, расте и количина времена потребна да се поменути алгоритми изврше. Зато, на пример, сви системи великих компанија зависе од алгоритама који су оптимизовани да нам резултате дају великом брзином.

Без одговарајуће анализе алгоритама, ови системи били би преспори, а осим тога и енергетски врло неефикасни, те прескупи. Овај проблем често решава посебна врста алгоритама који се заснивају на алгоритамској парадигми *подели па владај*, а математички модел који их прецизно објашњава јесу рекурентне релације.

1 Рекурентне релације

Рекурентне релације дефинишу низ у зависности од његових претходних чланова.

Дефиниција 1.1. Нека је $\{a_n\}_{n\in\mathbb{N}}$ низ реалних бројева. Рекурентна релација реда $k \in \mathbb{N}$ је једначина облика

$$a_n = f(n, a_{n-1}, a_{n-2}, \ldots, a_{n-k}) \quad \text{за} \quad n \geqslant k,$$

при чему је $f : \mathbb{N} \times X^k \to X$ функција која укључује k узастопних чланова низа, а X скуп коме припадају сви чланови низа.

Напомена 1.1. *Рекурентна релација је једнозначно одређена ако је задато k иницијалних вредности $a_0, a_1, \ldots, a_{k-1}$. Без задатих иницијалних вредности, релација описује бесконачну фамилију низова.*

Пример 1.1. Факторијел је дефинисан рекурентном релацијом

$$n! = n \cdot (n-1)! \quad \text{за} \quad n > 0$$

и иницијалном вредношћу $0! = 1$.

Пример 1.2. Фибоначијеви бројеви припадају низу чији је сваки члан збир претходна два члана. Фибоначијев низ је дефинисан рекурентном релацијом

$$F_n = F_{n-1} + F_{n-2}$$

и иницијалним вредностима $F_0 = 0$ и $F_1 = 1$. Првих неколико чланова Фибоначијевог низа јесу: $0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$

Дефиниција 1.2. Рекурентна релација је *линеарна* ако је

$$a_n = \sum_{i=1}^{k} c_i a_{n-i} + g(n),$$

при чему су c_i константе или функције од n, а $g(n)$ слободан члан.

Дефиниција 1.3. Линеарна рекурентна релација је *хомогена* ако је $g(n) = 0$, а иначе је *нехомогена*.

Лема 1.1 (Линеарност)**.** *Нека су $\{a_n\}$ и $\{b_n\}$ решења хомогене линеарне рекурентне релације $x_n = \sum_{i=1}^{k} c_i x_{n-i}$. Тада је за све константе $\alpha, \beta \in \mathbb{R}$ низ*

$$u_n = \alpha a_n + \beta b_n$$

такође решење исте рекурентне релације.

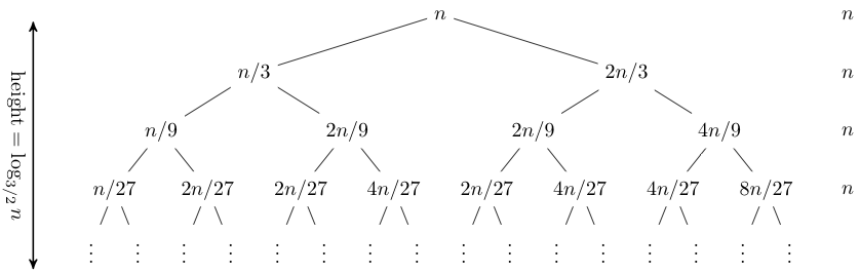
Доказ. Претпоставимо да су $\{a_n\}$ и $\{b_n\}$ решења релације $\{x_n\}$. То значи да се могу записати као $a_n = \sum_{i=1}^{k} c_i a_{n-i}$ и $b_n = \sum_{i=1}^{k} c_i b_{n-i}$. За $n \geqslant k$ и $\alpha, \beta \in \mathbb{R}$ дефинишимо нови низ

$$\begin{aligned} u_n = \alpha a_n + \beta b_n &= \alpha(c_1 a_{n-1} + \cdots + c_k a_{n-k}) + \beta(c_1 b_{n-1} + \cdots + c_k b_{n-k} \\ &= c_1(\alpha a_{n-1} + \beta b_{n-1}) + \cdots + c_k(\alpha a_{n-k} + \beta b_{n-k}) \\ &= c_1 u_{n-1} + c_2 u_{n-2} + \cdots + c_k u_{n-k}. \end{aligned}$$

Низ $\{u_n\}$ задовољава исту рекурентну релацију, те је и он њено решење. □

Решити рекурентну релацију значи наћи експлицитну формулу за општи члан низа. Неки од начина решавања су:

- итеративна метода (нпр. a_{n-1} се изрази преко a_{n-2});
- метода карактеристичне једначине (уз претпоставку $a_n = r^n$);
- помоћу мастер теореме (теорема 2.1) и
- помоћу рекурзивног стабла (слика 1).



Слика 1: Рекурзивно стабло

2 Алгоритамска парадигма „подели па владај”

2.1 Алгоритмика

Алгоритмика је област информатике која се бави анализом временске и просторне сложености програма, као и дизајном што ефикаснијих алгоритама који решавају неки проблем. *Алгоритамска сложеност* је функција која описује како ресурси које алгоритам троши зависе од величине улаза. Ресурси су углавном време извршавања, меморија и енергија. *Величина улаза* n је мера количине информација које алгоритам прима и најчешће представља број битова у броју, број елемената у низу, број чворова у графу или дужину ниске. *Асимптотска нотација* описује понашање функције када $n \to \infty$, занемарујући константе и ниже редове.

Дефиниција 2.1. Нека су f и g две позитивне функције од $n \in \mathbb{N}$. За функцију $g(n)$ каже се да је *асимптотска горња граница* функције $f(n)$ и пише се $f(n) = \mathcal{O}\big(g(n)\big)$ ако постоје позитивне константе c и n_0 такве да за свако $n > n_0$ важи $f(n) \leqslant cg(n)$.

Дефиниција 2.2. За функцију $g(n)$ каже се да је *асимптотска доња граница* функције $f(n)$ и пише се $f(n) = \Omega\big(g(n)\big)$ ако постоје позитивне константе c и n_0 такве да за свако $n > n_0$ важи $f(n) > cg(n)$.

Дефиниција 2.3. Ако за две функције f и g важи $f(n) = \mathcal{O}\big(g(n)\big)$ и $f(n) = \Omega\big(g(n)\big)$, онда за функцију $g(n)$ кажемо да је *асимптотска тесна граница* функције $f(n)$ и пишемо $f(n) = \Theta\big(g(n)\big)$.

Дефиниција 2.4. За $k > 0$, $0 < \varepsilon < 1$ и $c > 1$ дефинише се строго уређена хијерархија раста функција

$$1 \ll \log n \ll \log^k n \ll n^\varepsilon \ll n \ll n \log n \ll n^2 \ll n^3 \ll \cdots \ll c^n \ll n! \ll n^n.$$

2.2 О парадигми

Алгоритми засновани на алгоритамској парадигми *подели па владај* (енг. *divide and conquer*) решавају проблем тако што га поделе на мање потпроблеме исте врсте, а затим реше потпроблеме да би на крају спојили парцијална решења у решење проблема.

Дефиниција 2.5. Ако је величина проблема n, а алгоритам решава $a \geqslant 1$ потпроблема величине $\frac{n}{b}$ ($b > 1$ је фактор смањења величине проблема) уз додатни рад $f(n)$ на поделу и комбиновање, рекурентна релација која описује *подели па владај* алгоритме је

$$T(n) = aT\Big(\frac{n}{b}\Big) + f(n).$$

Асимптотско решење ове рекурентне релације даје теорема 2.1.

Теорема 2.1 (Мастер теорема)**.** *За рекурентну релацију* $T(n) = aT(\frac{n}{b}) + f(n)$ *и* $\lambda = \log_b a$ *важе следећа три случаја:*

1. *ако је* $f(n) = \mathcal{O}(n^{\lambda-\varepsilon})$*, тада је* $T(n) = \Theta(n^\lambda)$*;*
2. *ако је* $f(n) = \Theta(n^\lambda \log^k n)$*, тада је* $T(n) = \Theta(n^\lambda \log^{k+1} n)$*;*
3. *ако је* $f(n) = \Omega(n^{\lambda+\varepsilon})$ *и* $af(\frac{n}{b}) \leqslant cf(n)$ *за неко* $c < 1$*, тада је* $T(n) = \Theta\big(f(n)\big)$.

У табели 1 дато је поређење неких релевантних *подели па владај* алгоритама по временској и просторној сложености.

Алгоритам	**Рекурентна релација**	**Време**	**Простор**
Merge Sort	$T(n) = 2T(\frac{n}{2}) + n$	$\Theta(n \log n)$	$\Theta(n)$
Quick Sort	$T(n) = T(k) + T(n-k) + n$	$\Theta(n \log n)$	$\mathcal{O}(\log n)$
Binary Search	$T(n) = T(\frac{n}{2}) + 1$	$\Theta(\log n)$	$\mathcal{O}(1)$

Табела 1: Поређење неких релевантних алгоритама

Закључак

Велике заслуге за брзину којом данас извршавамо свакодневне активности припадају алгоритмима заснованим на парадигми *подели па владај* и математичком моделу на коме се темеље — рекурентним релацијама. У овом раду је систематично објашњено како теоријска сарадња низова и алгоритмике дају практично решење за временски, енергетски и финансијски ефикасније програме.