Rekurentne relacije

06.11.2024.

1 Rekurentne relacije

1.1 Definicija rekurentnih relacija

Definicija Rekurzivna funkcija je funkcija čiji je domen skup nenegativnih brojeva sa sledećim osobinama 1) $f(0)$ je dato definicijom 2) $f(n)$ je definisano kao funkcija od elemenata skupa $\{f(0), f(1), ..., f(n-2), f(n-1)\}$

Datu definiciju možemo prevesti na jezik nizova

Definicija Neka je $\{a_n\}$ niz. Rekurentna relacija niza $\{a_n\}$ je jednakost koji element a_n izražava u funkciji od $\{a_1, a_2, ..., a_n\}$. Niz koji zadovoljava rekurentnu relaciju nazivamo **rešenjem rekurentne relacije**

Većina problema vezanih za rekurentne relacije se svodi na pronalaženje njihovog rešenja

1.2 Generisanje rekurentne relacije

Primer Naći rekurentnu relaciju i dati početne uslove za broj nizova bitova dužine n koji nemaju dve uzastopne nule. Koliko ima takvih nizova bitova dužine pet?

Rešenje: Neka a_n označava broj nizova bita dužine n koji nemaju dve uzastopne nule. Da bismo dobili relaciju rekurzije za $\{a_n\}$, primećujemo da, prema pravilu zbira, broj nizova bita dužine n koji nemaju dve uzastopne nule jeste jednak broju takvih nizova koji se završavaju sa 0 plus broj takvih nizova koji se završavaju sa 1. Pretpostavljamo da je $n \geq 3$, tako da niz ima najmanje tri bita.

Nizovi bita dužine n koji se završavaju sa 1 i nemaju dve uzastopne nule su tačno nizovi dužine $n-1$ bez dve uzastopne nule sa dodatom 1 na kraju. Dakle, postoji a_{n-1} takvih nizova.

Nizovi bita dužine n koji se završavaju sa 0 i nemaju dve uzastopne nule moraju imati 1 kao $(n-1)$-i bit; inače bi završavali sa parom nula. Iz toga sledi da su nizovi bita dužine n koji se završavaju sa 0 i nemaju dve uzastopne nule tačno nizovi dužine $n-2$ bez dve uzastopne nule sa dodatkom 10 na kraju. Dakle, postoji a_{n-2} takvih nizova.

Zaključujemo da:

$a_n = a_{n-1} + a_{n-2}$, za $n \geq 3$.

Početni uslovi su $a_1 = 2$, jer oba niza bitova dužine jedan, 0 i 1, nemaju uzastopne nule, i $a_2 = 3$, jer su validni nizovi bita dužine dva: 01, 10 i 11. Da bismo izračunali a_5, koristimo relaciju rekurzije tri puta:

$a_3 = a_2 + a_1 = 3 + 2 = 5$,
$a_4 = a_3 + a_2 = 5 + 3 = 8$,
$a_5 = a_4 + a_3 = 8 + 5 = 13$.

1.3 Rešavanje rekurentnih jednačina

U opštem slučaju, rekurentne jednačine se rešavaju pomoću matematičke indukcije, ali je ovakvo rešavanje često neintuitivno i zahteva uočavanje obrasca koji može biti veoma složen

Primer Fibonačijevi brojevi su niz definisan na sledeći način: $a_0 = 0$, $a_1 = 1$, $a_n = a_{n-1} + a_{n-2}$. Rešenje njihove rekurentne jednačine je niz definisan zatvorenom formom

$$a_n = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n)$$

Srećom po nas, prethodna rekurentna jednačina spada u vrstu onih čije rešenje možemo naći jednostavnim postupkom

1.4 Linearne rekurentne relacije sa konstantnim koeficijentima

Definicija Linearna rekurentna relacija sa konstantnim koeficijentima je rekurentna jednačina oblika

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k} + f(n)$$

gde su $c_1, \dots, c_k \in R$, $c_k \neq 0$ (tj. jednačina je reda k).

Rekurentna relacija	Linearna	Homogena	Konstantni koeficijenti	Objašnjenje
$a_n = 2a_{n-1} + 3a_{n-2}$	Da	Da	Da	Linearna je, homogena je jer nema dodatnih članova (sve zavisi samo od prethodnih članova), ima konstantne koeficijente.
$a_n = a_{n-1}^2 + 2$	Ne	Ne	Da	Nije linearna jer ima kvadratni član a_{n-1}^2, i nije homogena zbog dodatnog člana 2.
$a_n = 3a_{n-1} + 5n$	Da	Ne	Da	Linearna je jer je stepen svakog člana 1, ali nije homogena zbog člana $5n$ koji zavisi od n.
$a_n = n \cdot a_{n-1} + 4$	Da	Ne	Ne	Linearna je, ali nije homogena zbog člana 4 i nema konstantne koeficijente jer n menja koeficijent a_{n-1}.
$a_n = 4a_{n-1} - 2a_{n-2} + 5$	Da	Ne	Da	Linearna je i ima konstantne koeficijente, ali nije homogena zbog dodatnog člana 5.
$a_n = a_{n-1} + a_{n-2} + n \cdot a_{n-3}$	Da	Da	Ne	Linearna i homogena je jer nema dodatnih članova, ali nema konstantne koeficijente zbog $n \cdot a_{n-3}$.
$a_n = 3a_{n-1} + 2a_{n-2} - 4a_{n-3}$	Da	Da	Da	Linearna, homogena i ima konstantne koeficijente.
$a_n = 2a_{n-1} + 3a_{n-2} + 7n^2$	Da	Ne	Da	Linearna je, ali nije homogena zbog člana $7n^2$.
$a_n = 5n \cdot a_{n-1} - 3$	Da	Ne	Ne	Linearna je, nije homogena zbog -3, i nema konstantne koeficijente zbog $5n \cdot a_{n-1}$.

Rekurentna relacija	Linearna	Homogena	Konstantni koeficijenti	Objašnjenje
$a_n = a_{n-1}^2 - a_{n-2} + 1$	Ne	Ne	Da	Nije linearna zbog kvadrata a_{n-1}^2 i nije homogena zbog dodatnog člana 1.

1.4.1 Linearne homogene rekurentne jednačine sa konstantnim koeficijentima

Definicija Linearna homogena rekurentna jednačina sa konstantnim koeficijentima je rekurentna jednačina oblika

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k},$$

tj. jednačina gde je $f(n) = 0$ i gde su c_i konstante, a a_{n-i} prethodni članovi niza $(1 \leq i \leq k)$

Pretpostavimo da postoji broj x sa osobinom da je $a_i = x^i$. Zamenom u rekurentnu jednačinu dobijamo

$$x^n = c_1 x^{n-1} + c_2 x^{n-2} + ... + c_k x^{n-k}$$

Skraćivanjem x^{n-k} sa obe strane i prebacivanjem svih izraza na jednu stranu dobijamo

$$x^k - c_1 x^{k-1} - c_2 x^{k-2} - ... c_{k+1} x - c_k = 0$$

Data jednačina naziva se **karakteristična jednačina rekurentne relacije**

Teorema Ukoliko su $x_1, x_2, ..., x_k$ koreni karakteristične jednačine rekurentne jednačine $a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$ i ukoliko je mnogostrukost svakog korena tačno jedan onda je opšte rešenje dato sa

$$a_n = b_1 x_1^n + b_2 x_2^n + ... + b_k x_k^n$$

gde su b_i konstante jedinstveno određene početnim uslovima

Dokaz Tačnost formule za a_n se dokazuje jednostavnom zamenom u rekurentnu jednačinu. Jedinstvenost koeficijenata sledi iz činjenice da se zamenom u početne uslove dobija sistem

$$a_0 = b_1 + b_2 + ... + b_k$$

$$a_1 = b_1 x_1 + b_2 x_2 + ... + b_k x_k$$

$$a_2 = b_1 x_1^2 + b_2 x_2^2 + ... + b_k x_k^2$$

$$...$$

$$a_{k-1} = b_1 x_1^{k-1} + b_2 x_2^{k-1} + ... + b_k x_k^{k-1}$$

Determinanta ovog sistema jednaka je determinanti Vandermondove matrice, koja je u ovom slučaju različita od nule, jer su svi brojevi $x_1, x_2, ..., x_n$ različiti

Korišćenjem date teoreme, možemo izvesti zatvorenu formu jednačine Fibonačijevih brojeva!

Znamo da su $a_0 = 0$ i $a_1 = 1$. Karakteristična jednačina rekurentne relacije $a_n = a_{n-1} + a_{n-2}$ jednaka je $x^2 - x - 1 = 0$. Rešenja date jednačine su $\frac{1+\sqrt{5}}{2}$ i $\frac{1-\sqrt{5}}{2}$. Dakle, znamo da je zatvorena forma oblika

$$a_n = b_1 (\frac{1+\sqrt{5}}{2})^n + b_2 (\frac{1-\sqrt{5}}{2})^n$$

Zamenom $n = 0$ i $n = 1$ u datu formulu dobijamo sistem jednačina

$$0 = b_1 + b_2$$

$$1 = b_1 \frac{1+\sqrt{5}}{2} + b_2 \frac{1-\sqrt{5}}{2}$$

Rešavanjem datog sistema dobijamo vrednosti $b_1 = \frac{1}{\sqrt{5}}$ i $b_2 = -\frac{1}{\sqrt{5}}$ i zamenom u izraz za a_n dobijamo rešenje rekurentne jednačine

$$a_n = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n)$$

Teorema Ukoliko su $x_1, x_2, \dots, x_r$ koreni karakteristične jednačine rekurentne jednačine $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$, gde koren x_i ima mnogostrukost m_i (za $i = 1, 2, \dots, r$), tada je opšte rešenje dato sa

$$a_n = \sum_{i=1}^{r} \left(b_{i,1} x_i^n + b_{i,2} n x_i^n + b_{i,3} n^2 x_i^n + \cdots + b_{i,m_i} n^{m_i - 1} x_i^n \right),$$

gde su $b_{i,j}$ konstante koje su jedinstveno određene početnim uslovima.

Dokaz Slično prethodnom dokazu.

1.4.2 Linearne nehomogene rekurentne jednačine sa konstantnim koeficijentima

Definicija Linearna nehomogena rekurentna jednačina sa konstantnim koeficijentima je rekurentna jednačina oblika

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + f(n),$$

tj. jednačina gde je $f(n) \neq 0$ i gde su c_i konstante, a a_{n-i} prethodni članovi niza $(1 \leq i \leq k)$

Teorema Ako je $a_n^{(p1)}$ partikularno rešenje nehomogene linearne rekurentne relacije sa konstantnim koeficijentima, tada za svako rešenje a_n jednačine iz prethodne definicije postoji rešenje $a_n^{(h)}$ odgovarajuće homogene rekurentne relacije sa osobinom

$$a_n = a_n^{(h)} + a_n^{(p1)}, \quad n \geq 0.$$

Dokaz. Neka je $a_n^{(p1)}$ dato (partikularno) rešenje jednačine. Tada za svako $n \in \mathbb{N}_0$ važi

$$a_n^{(p1)} = c_1 a_{n-1}^{(p1)} + c_2 a_{n-2}^{(p1)} + \cdots + c_k a_{n-k}^{(p1)} + f(n). \tag{1.1}$$

Posmatrajmo sada proizvoljno rešenje $a_n^{(p)}$ jednačine. Za njega važi

$$a_n^{(p)} = c_1 a_{n-1}^{(p)} + c_2 a_{n-2}^{(p)} + \cdots + c_k a_{n-k}^{(p)} + f(n), \quad n \in \mathbb{N}_0. \tag{1.2}$$

Ako od relacije (1.2) oduzmemo relaciju (1.1), dobijamo

$$a_n^{(p)} - a_n^{(p1)} = c_1 \left(a_{n-1}^{(p)} - a_{n-1}^{(p1)} \right) + c_2 \left(a_{n-2}^{(p)} - a_{n-2}^{(p1)} \right) + \cdots + c_k \left(a_{n-k}^{(p)} - a_{n-k}^{(p1)} \right),$$

što znači da je $a_n^{(p)} - a_n^{(p1)}$ rešenje homogenog dela rekurentne relacije, tj. za svako rešenje nehomogene rekurentne relacije a_n postoji rešenje $a_n^{(h)}$ homogene jednačine tako da se a_n može izraziti pomoću $a_n^{(h)}$ i $a_n^{(p1)}$ koristeći obrazac

$$a_n = a_n^{(h)} + a_n^{(p1)}.$$

Teorema Neka je

$$f(n) = \left(b_m n^m + b_{m-1} n^{m-1} + \cdots + b_1 n + b_0\right) s^n, \quad b_0, \ldots, b_m \in \mathbb{R}.$$

Ako je s koren karakteristične jednačine višestrukosti l (ako nije koren, tada je $l = 0$), onda postoji partikularno rešenje oblika

$$a_n^{(p)} = n^l \left(c_m n^m + c_{m-1} n^{m-1} + \cdots + c_1 n + c_0\right) s^n.$$

Teorema Ako su $a_n^{(p1)}$ i $a_n^{(p2)}$ redom rešenja nehomogenih rekurentnih relacija

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_1(n)$$

i

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_2(n),$$

onda je $a_n^{(p1)} + a_n^{(p2)}$ rešenje nehomogene rekurentne relacije

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_1(n) + f_2(n).$$

Dokaz Ako su $a_n^{(p1)}$ i $a_n^{(p2)}$ redom rešenja navedenih rekurentnih relacija, onda važi

$$a_n^{(p1)} = c_1 a_{n-1}^{(p1)} + c_2 a_{n-2}^{(p1)} + \cdots + c_k a_{n-k}^{(p1)} + f_1(n)$$

i

$$a_n^{(p2)} = c_1 a_{n-1}^{(p2)} + c_2 a_{n-2}^{(p2)} + \cdots + c_k a_{n-k}^{(p2)} + f_2(n).$$

Sabiranjem prethodne dve jednakosti dobijamo

$$a_n^{(p1)} + a_n^{(p2)} = c_1 \left(a_{n-1}^{(p1)} + a_{n-1}^{(p2)}\right) + c_2 \left(a_{n-2}^{(p1)} + a_{n-2}^{(p2)}\right) + \cdots + c_k \left(a_{n-k}^{(p1)} + a_{n-k}^{(p2)}\right) + f_1(n) + f_2(n),$$

čime je direktno pokazano da je $a_n^{(p1)} + a_n^{(p2)}$ rešenje rekurentne relacije

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_1(n) + f_2(n).$$

1.5 Primene u programiranju

[1]:
```
#Merge sort
def merge(arr, left, mid, right):
    n1 = mid - left + 1
    n2 = right - mid

    L = [0] * n1
    R = [0] * n2

    for i in range(n1):
        L[i] = arr[left + i]
    for j in range(n2):
        R[j] = arr[mid + 1 + j]

    i = 0
    j = 0
    k = left

    while i < n1 and j < n2:
        if L[i] <= R[j]:
            arr[k] = L[i]
            i += 1
        else:
            arr[k] = R[j]
            j += 1
        k += 1

    while i < n1:
        arr[k] = L[i]
        i += 1
        k += 1

    while j < n2:
        arr[k] = R[j]
        j += 1
        k += 1

def merge_sort(arr, left, right):
    if left < right:
        mid = (left + right) // 2

        merge_sort(arr, left, mid)
```

```
        merge_sort(arr, mid + 1, right)
        merge(arr, left, mid, right)

def print_list(arr):
    for i in arr:
        print(i, end=" ")
    print()

if __name__ == "__main__":
    arr = [12, 11, 13, 5, 6, 7, 24, 56, 12, 9, 80, 45]
    print("Početni niz: ")
    print_list(arr)

    merge_sort(arr, 0, len(arr) - 1)

    print("\nSortirani niz: ")
    print_list(arr)
```

```
Početni niz:
12 11 13 5 6 7 24 56 12 9 80 45

Sortirani niz:
5 6 7 9 11 12 12 13 24 45 56 80
```

[4]:
```
#Hanojeva kula
def TowerOfHanoi(n, from_rod, to_rod, aux_rod):
    if n == 0:
        return
    TowerOfHanoi(n-1, from_rod, aux_rod, to_rod)
    print("Pomeri disk", n, "sa diska", from_rod, "na disk", to_rod)
    TowerOfHanoi(n-1, aux_rod, to_rod, from_rod)

if __name__ == "__main__":
  N = 3
  TowerOfHanoi(N, 'A', 'C', 'B')
```

```
Pomeri disk 1 sa diska A na disk C
Pomeri disk 2 sa diska A na disk B
Pomeri disk 1 sa diska C na disk B
Pomeri disk 3 sa diska A na disk C
Pomeri disk 1 sa diska B na disk A
Pomeri disk 2 sa diska B na disk C
Pomeri disk 1 sa diska A na disk C
```

[1]:
```
#Binarna pretraga
def binarySearch(arr, low, high, x):
    while low <= high:
        mid = low + (high - low) // 2
```

```python
        if arr[mid] == x:
            return mid
        elif arr[mid] < x:
            low = mid + 1
        else:
            high = mid - 1
    return -1


if __name__ == '__main__':
    arr = [2, 3, 4, 10, 40]
    x = 10

    result = binarySearch(arr, 0, len(arr)-1, x)
    if result != -1:
        print(f"Element {x} se nalazi na indeksu: {result}")
    else:
        print(f"Element {x} ne postoji u nizu.")
```

```
Element 10 se nalazi na indeksu: 3
```