

# ფენჯიკის ხე



# ფენვიკის ხე (ორობითი ინდექს-ხე): ამოცანის დასმა

ვთქვათ, მოცემულია  $n$  ცალი რიცხვისაგან  
შედგენილი მიმდევრობა

- 1.) გავზარდოთ  $i$ -ური წევრის მნიშვნელობა.
- 2.) შეკითხვა: დავადგინოთ ელემენტთა ჯამის მნიშვნელობა  $[k, j]$  ინტერვალში

ფენვიკის ხის დადებითი მხარე:

- შეკითხვა შესრულებას უარეს შემთხვევაში სჭირდება  $O(\log n)$  დრო.
- მოკლე კოდი.

# ამოხსნა სტატიკური მონაცემებისათვის

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ელემენტი	1	0	2	1	1	3	0	4	2	5	2	2	3	1	0	2
ჯამი	1	1	3	4	5	8	8	12	14	19	21	23	26	27	27	29

რაიმე  $B$  მასივში  $i$ -ურ ინდექსზე შევინახოთ 1-დან  $i$ -მდე ელემენტების ჯამი, მაშინ  $[l, r]$  ინტერვალში მყოფი ელემენტების ჯამი ტოლი იქნება:  $B[r]-B[l]$ .

# ფენვიკის ხე: შესავალი

- ჩვენი მიზანია გამოვთვალოთ ელემენტთა ჯამი  $[1, i]$  ინტერვალში.
- გამოვიყენოთ ის ფაქტი, რომ ნებისმიერი რიცხვი შეიძლება წარმოვადგინოთ 2-ის ხარისხების ჯამით.
- გამოვიყენოთ ეს თვისება  $[1, i]$  ინტერვალის წარმოსადგენად.
- $13 = 8 + 4 + 1$
- $[1, 13] = [1, 8] + [9, 12] + [13, 13]$

# ფენვიკის ხე: ინტერვალები

- ხის წვეროებში ჩვენ ვინახავთ  $[i - 2^r + 1, i]$  ინტერვალში შემავალი ელემენტების ჯამს, სადაც  $r$  არის ბოლო არანულოვანი ციფრის პოზიცია  $i$ -ის ორობით ჩანაწერში.

- მაგალითი:

$13_{10} = 1101_2$ , ბოლო არანულოვანი ციფრი დგას 0 პოზიციაზე.

$4_{10} = 100_2$ , ბოლო არანულოვანი ციფრი დგას 2 პოზიციაზე.

# ფენვიკის ხის სტრუქტურა

$f[i] = i$ -ური ელემენტის მნიშვნელობა

$c[i] = f[1] + f[2] + \dots + f[i]$

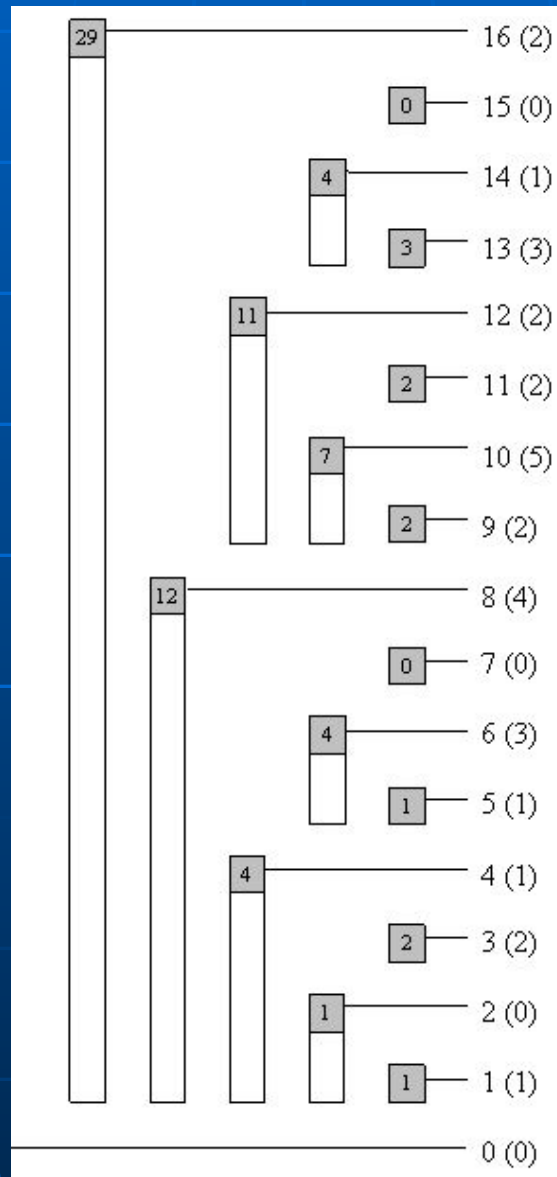
$tree[i] = [i - 2^r + 1, i]$  ინტერვალში  
მყოფი ელემენტების ჯამი

$i$	$(i - 2^r + 1) \dots i$
1	1
2	1...2
3	3
4	1...4
5	5
6	5...6
7	7
8	1...8
9	9
10	9...10
11	11
12	9...12
13	13
14	13...14
15	15
16	1...16

# ფენვიკის ხის მაგალითი

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f	1	0	2	1	1	3	0	4	2	5	2	2	3	1	0	2
c	1	1	3	4	5	8	8	12	14	19	21	23	26	27	27	29
tree	1	1	2	4	1	4	0	12	2	7	2	11	3	4	0	29

# ფენვიკის ხის სტრუქტურა





# ფუნქციის ხე: შეკითხვა (Query)

```
int read(int idx)
{
    int sum = 0;
    while (idx > 0)
    {
        sum += tree[idx];
        idx = idx - (idx & -idx);
    }
    return sum;
}
```

როგორ მიუძღობს  $idx -= (idx \& -idx)$ ?

$$idx = 44 = 101100_2$$

**Sum = tree[44];**

$$-idx = 010011+1=010100_2$$

$$101100_2 \& 010100_2 = 000100_2 = 4$$

$$idx = 44-4 = 40 = 101000_2 \quad \text{Sum = tree[44]+tree[40];}$$

$$-idx = 010111+1=011000_2$$

$$101000_2 \& 011000_2 = 001000_2 = 8$$

$$idx = 40-8 = 32 = 100000_2 \quad \text{Sum = tree[44]+tree[40]+tree[32];}$$

$$-idx = 011111+1=100000_2$$

$$idx = 32-32 = 0$$

**Sum = tree[44]+tree[40]+tree[32];**

# ფენვიკის ხე: განახლება (Update)

```
void update(int idx ,int val)
{
    while (idx <= MaxVal)
    {
        tree[idx] += val;
        idx += (idx & -idx);
    }
}
```

# ასიმპტოტიკა

როგორც განახლების, ასევე შეკითხვის  
დამუშავების დროა  $O(\log(n))$ .