

პრაქტიკული მეცადინება 6

სააუდიტორო სამუშაო:

- ზოგადი ცნობები
- ამოცანა /გროვის ძირითადი თვისების აღდგენა/ >>>
- სავარჯიშოები >>>

ზოგადი ცნობები

ვითყვი, რომ სწრაფი წვდომის იტერატორების $[a, b)$ დიაპაზონი წარმოადგენს ბინარულ გროვას, თუ თუ ყოველი $0 < i < n$ ინდექსისთვის სრულდება **max-გროვის ძირითადი თვისება** (heap property):

$$a[\text{parent}(i)] \geq a[i],$$

სადაც $n = b - a$, ხოლო $\text{int PARENT}(i) \{ \text{return } \lfloor (i-1)/2 \rfloor; \}$.

$\text{PARENT}(i)$ -ის გარდა, გამოვიყენებთ კიდევ ორ ფუნქციას:

```
int LEFT(i) {return 2*i + 1;} და int RIGHT(i) {return 2*(i+1);}
```

რომლებსაც თვალსაჩინო გეომეტრიული შინაარსი გააჩნია, თუ გროვას წარმოვადგენთ ბინარული ხის სახით. მეხსიერებაში გროვა წარმოადგენს კერძოდ ორგანიზებულ კონტეინერის ფრაგმენტს, ხის სახით მისი წარმოდგენა მხოლოდ იმისთვის არის საჭირო რომ ადვილად გავიაზროთ გროვის ალგორითმების მოქმედება.

ბინარული გროვის აგება შეიძლება ორნაირად. შეიძლება თითო-თითოდ დავამატოთ ელემენტები გროვას `pop_heap` ალგორითმით, ვიდრე საჭირო ფრაგმენტს მთლიანად არ ვაქცევთ გროვად. მაგრამ ასეთი ალგორითმის შესრულების დრო არის $n \log n$. შესაძლებელია გამოვიყენოთ ძალიან მარტივი ალგორითმი, რომელიც სრულდება წრფივ დროში:

```
make_heap(a,b)
1 | n = b - a;
2 | for (i = (n-2)/2; i >= 0; --i)
3 |     heapify(a,b,i);
```

რომელიც დაიწყება ბოლოდან, შექმნის ორელემენტოვან გროვებს, ორელემენტოვანებიდან ააწყობს სამელემენტოვანებს და ა. შ. ვიდრე მიტიტებულ დიაპაზონს არ აქცევს გროვად.

`heapify(a,b,i)`-ს ეწოდება გროვის თვისების აღდგენის ალგორითმი. მისი ფსევდოკოდი არის:

```
heapify(a,b,i)
1 | l = LEFT(i); r = RIGHT(i);
2 | if (l < b - a && a[l] > a[i])
3 |     largest = l;
4 | else largest = i;
5 | if (r < b - a && a[r] > a[largest])
6 |     largest = r;
7 | if (largest != i)
8 |     ადგილები გაუცვალეთ a[i] და a[largest] ელემენტებს;
9 |     HEAPIFY(a,b, largest);
```

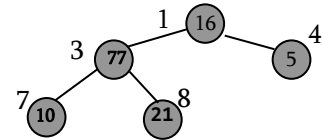
იმისათვის, რომ ამ ალგორითმმა კორექტული შედეგი მოგვცეს, ქვეხეები წვეროებით `LEFT(i)` და `RIGHT(i)` უნდა წარმოადგენდნენ გროვას.

=== ამოცანა. მოცემულია მთელი რიცხვების a მასივის ფრაგმენტი. $[a, a+9]$ დიაპაზონის $i=1$ წვეროში ადადგინეთ გროვის თვისება, შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით:

i	0	1	2	3	4	5	6	7	8
a[i]	10	16	100	77	5	50	80	10	21

რა პირობა არის აუცილებელი იმისათვის, რომ $\text{heapify}(a, a+9, i)$ კორექტულად შესრულდეს?

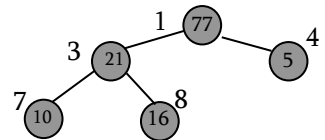
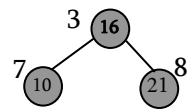
ამოხსნა. გამოვიყენოთ ზემოთ მოყვანილი ფსევდოკოდი. თვალსაჩინოებისთვის, დავხატოთ ქვეზე ფსევით მეორე წვეროში:



```

heapify(a, a+9, 1)
| n = 9; // დიაპაზონში ელემენტების რაოდენობა
| l = 3; r = 4;
| if (3 < 9 && 77 > 16); ☹ // შესრულდა მესამე სტრიქონის if შეტყობინების პირობა
|   largest = 3; // ჯერჯერობით, უდიდესი მნიშვნელობა მეოთხე წვეროშია
| if (4 < 9 && 5 > 77); ☹ // if-ის პირობა არ შესრულდა, ამიტომ largest არ შეიცვლება;
| if (1 != 3) ☹ // პირობა largest != i შესრულდა
|   | 16 ⇌ 77; // კონტეინერის მე-2 და მე-4 უჯრებში გადავცვლით მნ-ბებს
heapify(a, a+9, 3)
| l = 7; r = 8;
| if (7 < 9 && 10 > 16); ☹ // if შეტყობინების პირობა არ შესრულდა
| else largest = 3; // უდიდესი მნიშვნელობა ისევ მეოთხე წვეროშია
| if (8 < 9 && 21 > 16); ☹ // შესრულდა მესამე სტრიქონის if შეტყობინების პირობა
|   | largest = 8; // უდიდესი მნიშვნელობა ისევ მეცხრე წვეროშია
|   if (3 != 8) ☹ // პირობა შესრულდა
|     | 16 ⇌ 21;
heapify(a, a+9, 8)
| l = 17, r = 18;
| if (17 < 9 ...); ☹ // პირობა არ შესრულდა,
|   | largest = 8;
| if (18 < 9 ...); ☹ // if შეტყობინების პირობა არ შესრულდა,
|   if (8 != 8) ☹ // if შეტყობინების პირობა არ შესრულდა და დამთავრდა.

```



8-ე სტრიქონის

8 | ადგილები გავუცვალეთ $a[i]$ და $a[largest]$ ელემენტებს;

ყოველი შესრულების შემდეგ, როგორც ხის ფრაგმენტზე, ასევე კონტეინრის დიაპაზონზე უნდა ავსახოთ შესაბამისი ცვლილებები. მაგალითად, ალგორიტმის შესრულების შემდეგ კონტეინერში ჩაწერილი იქნება:

i	0	1	2	3	4	5	6	7	8
a[i]	10	77	100	21	5	50	80	10	16

<<< სავარჯიშოები.

1. მთელი რიცხვების მასივის მოცემული დიაპაზონისთვის

i	0	1	2	3	4	5	6	7	8	9
a[i]	10	16	10	77	5	50	8	10	21	4

a) i=2 წვეროში

b) i=1 წვეროში

აღადგინეთ გროვის ძირითადი თვისება, შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით:

2. მთელი რიცხვების მასივის მოცემული დიაპაზონისთვის i=1 წვეროში აღადგინეთ გროვის ძირითადი თვისება, შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით:

i	0	1	2	3	4	5	6	7	8
a[i]	10	16	10	77	5	5	80	10	21

a. მთელი რიცხვების a მასივის [a, a+7] დიაპაზონის i=1 წვეროში აღადგინეთ გროვის ძირითადი თვისება, შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით:

i	0	1	2	3	4	5	6	7	8	9
a[i]	10	16	10	5	55	4	8	10	21	4