პრაქტიკული 7: გროვის აგება

სააუდიტორიო სამუშაო:

- ზოგადი ცნობები
- ამოცანა /გროვის აგება/ >>>
- სავარჯიშოები >>>

ზოგადი ცნობები. გროვის აგების ალგორითმის ფსევდოკოდია:

```
MAKE_HEAP(a,b)
1   n = b - a;
2   for (i = (n-2)/2; i >= 0; --i)
3       HEAPIFY(a,b,i);
```

შემდეგ ამოცანაში 9 ელემენტიან ფრაგმენტს გარდავქმნით გროვად. ფსევდოკოდს აქვს სახე:

```
MAKE_HEAP(a,a+9)
1   n = 9;
2   for (i = 3 ; i >= 0; i--)
3       HEAPIFY(a,a+9,i);
```

<<< **ამოცანა.** *a* მასივის დასაწყისში წერია შემდეგი მთელი რიცხვები:

i	0	1	2	3	4	5	6	7	8	9	10	...
a[i]	12	40	15	67	3	24	31	10	16	77	13	...

მასივის `[a,a+9)` ფრაგმენტი გარდაქმენით გროვად. გროვის აგება აჩვენეთ ფსევდოკოდის ძირითადი შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით.

ამოხსნა. გროვის აგების `MAKE_HEAP(a,a+9)` ალგორითმში სტრიქონების მიმდევრობა (ტრასირება) ასეთია:

```
i = 3:
HEAPIFY(a,a+9,3)
    l = 7 ;  r = 8;
    if (7<9  &&  10>67)  ☹ ⇒ largest = 3;          //შესრულდა else შეტყობინება
    if (8<9  &&  16>67)  ☹ ⇒ largest არ შეცვლილა;
    if (3 != 3)  ☹ □                               // largest = i
i=2:
HEAPIFY(a,a+9,2)
    l = 5 ; r = 6;
    if (5 <9  &&  24>15) ☺⇒ largest = 5;          //შესრულდა if შეტყობინება
    if (6 <9  &&  31>24); ☺⇒ largest = 6;          //შესრულდა  if შეტყობინება
    if (2 != 6)  ☺                                 // largest != i
       15 ⇄ 31      // a[2] ⇄ a[6];                //ცვლილებები უნდა აისახოს კონტეინერში
       HEAPIFY(a,a+9,6)
            l = 13 , r = 14;
            if (13  < 9  ...);  ☹ ⇒ largest=6;     //შესრულდა else შეტყობინება
            if (14  < 9  ...);  ☹ ⇒ largest არ შეცვლილა;
            if (6 != 6)  ☹ □                       //a[6] ფოთოლია
i=1:
HEAPIFY(a,a+9,1)
    l = 3 ;  r = 4;
    if (3 < 9  && 67 > 40); ☺ ⇒ largest = 3;       //შესრულდა if შეტყობინება
    if (4 < 9  && 3>67);  ☹ ⇒ largest არ შეცვლილა;
    if (1 != 3) ☺                                  //largest  !=  i
```

40 ⇄ 67 //a[1] ⇄ a[3]; //ცვლილებები უნდა აისახოს კონტეინერში

HEAPIFY(a,a+9,3)

l = 7; r = 8;

if (7 < 9 && 10>40); ☹ ⇒ largest = 3; //შესრულდა else შეტყობინება

if (8 < 9 && 16>40); ☹⇒ largest არ შეცვლილა;

if (3!=3) ☹ □ //largest=i

i=0:

HEAPIFY(a,a+9,0)

l = 1; r = 2;

if (1 < 9 && 67 > 12); ☺ ⇒ largest = 1;

if (2 < 9 && 31 > 67); ☹⇒ largest არ შეცვლილა;

if (0 != 1) ☺ //largest != i

12 ⇄ 67 //a[0] ⇄ a[1];

HEAPIFY(a,a+9,1)

l = 3 ,r = 4

if (3 < 9 && 40>12); ☺⇒ largest = 3;

if (4 < 9 && 3>40); ☹ ⇒ largest არ შეცვლილა;

if (1 != 3) ☺ //largest != i

12 ⇄ 40 //ცვლილებები უნდა აისახოს კონტეინერში

HEAPIFY(a,a+9,**3**)

l = 7; r = 8;

if (7 < 9 && 10>12); ☹⇒ largest = 3; //შესრულდა else შეტყობინება

if (8 < 9 && 16>12); ☺ ⇒ largest = **8**;

if (3 != 8) ☺ //largest ≠ i

12 ⇄ 16 //ცვლილებები უნდა აისახოს კონტეინერში

HEAPIFY(a,a+9,8)

l = 17; r = 18;

if (17 <9 ...); ☹ ⇒ largest = 8; //შესრულდა else შეტყობინება

if (18 <9 ...); ☹⇒ largest არ შეცვლილა;

if (8!=8) ☹ □ //a[8] ფოთოლია

<<< სავარჯიშოები:

1.

i	0	1	2	3	4	5	6	7	8	9	10
a[i]	1	8	15	7	3	24	31	10	16	7	13

კონტეინერის [&a[0], &a[8]) ფრაგმენტისგან ააგეთ გროვა ფსევდოკოდის ძირითადი შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით.

2.

i	0	1	2	3	4	5	6	7	8	9	10
a[i]	12	40	15	17	83	24	31	10	16	7	93

კონტეინერის [&a[0], &a[9]) ფრაგმენტისგან ააგეთ გროვა ფსევდოკოდის ძირითადი შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით.

3.

i	0	1	2	3	4	5	6	7	8	9	10
a[i]	12	40	15	67	3	4	31	10	16	71	13

კონტეინერის [&a[0], &a[10]) ფრაგმენტისგან ააგეთ გროვა ფსევდოკოდის ძირითადი შეტყობინებების (ანუ სტრიქონების) მიმდევრობის მითითებით.