

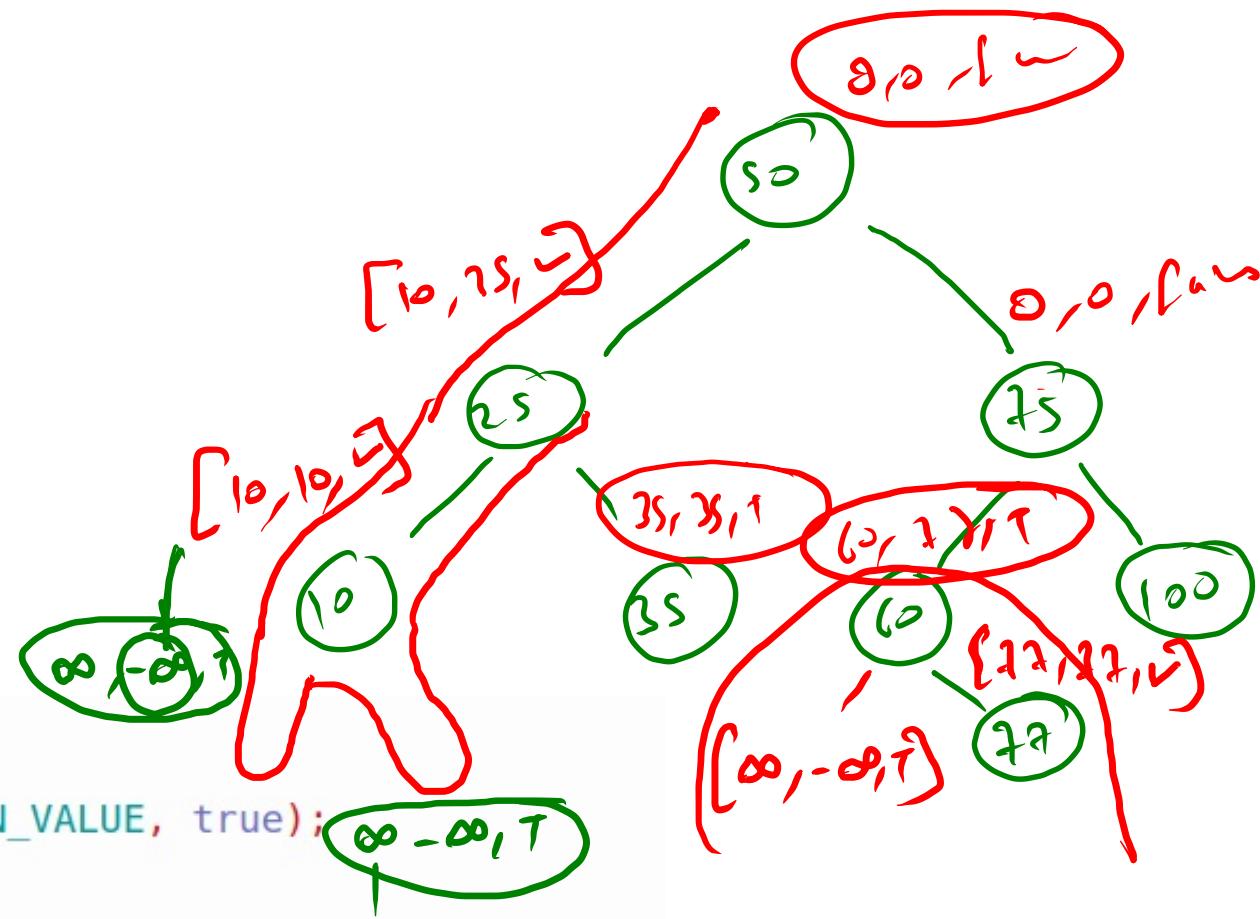
```

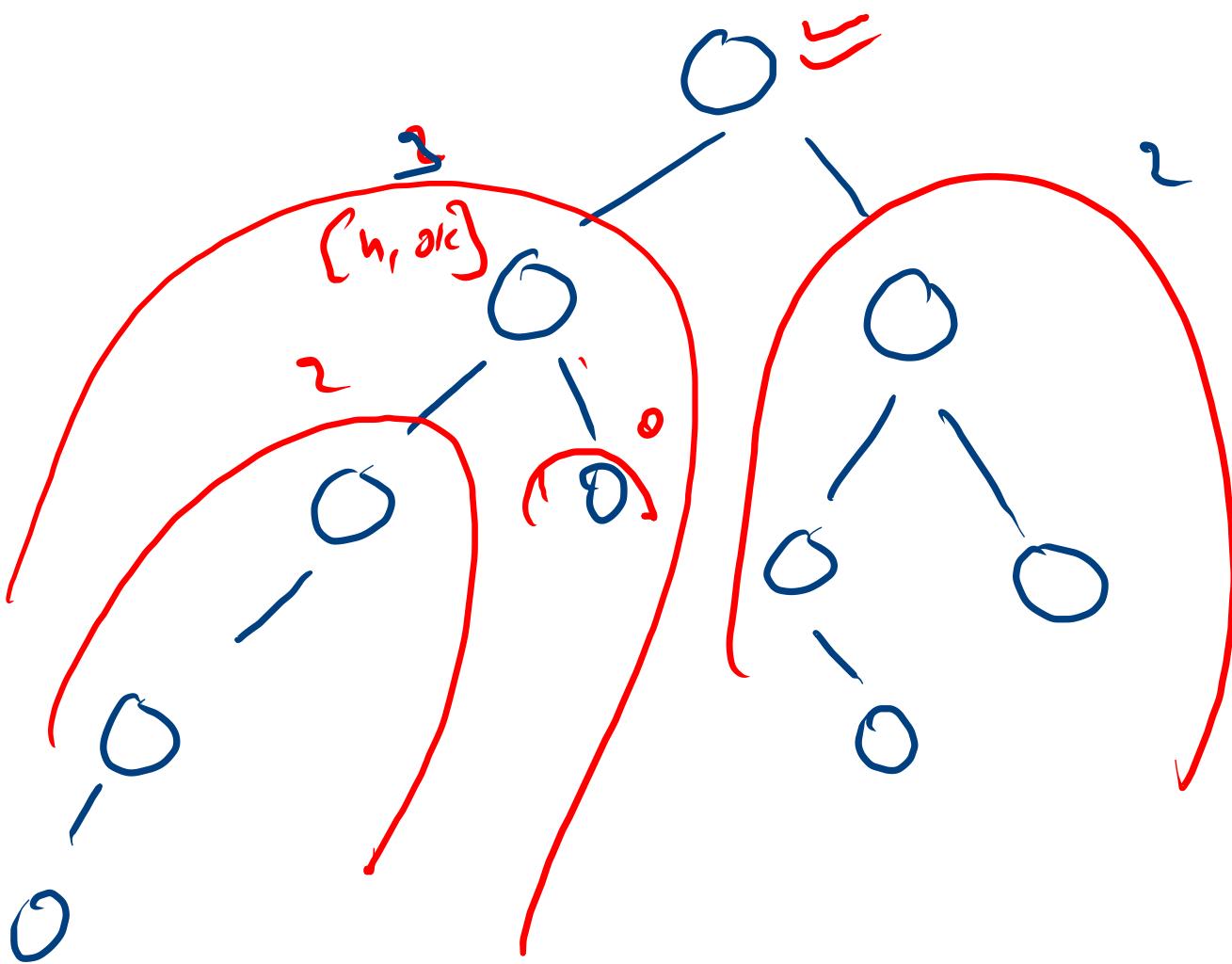
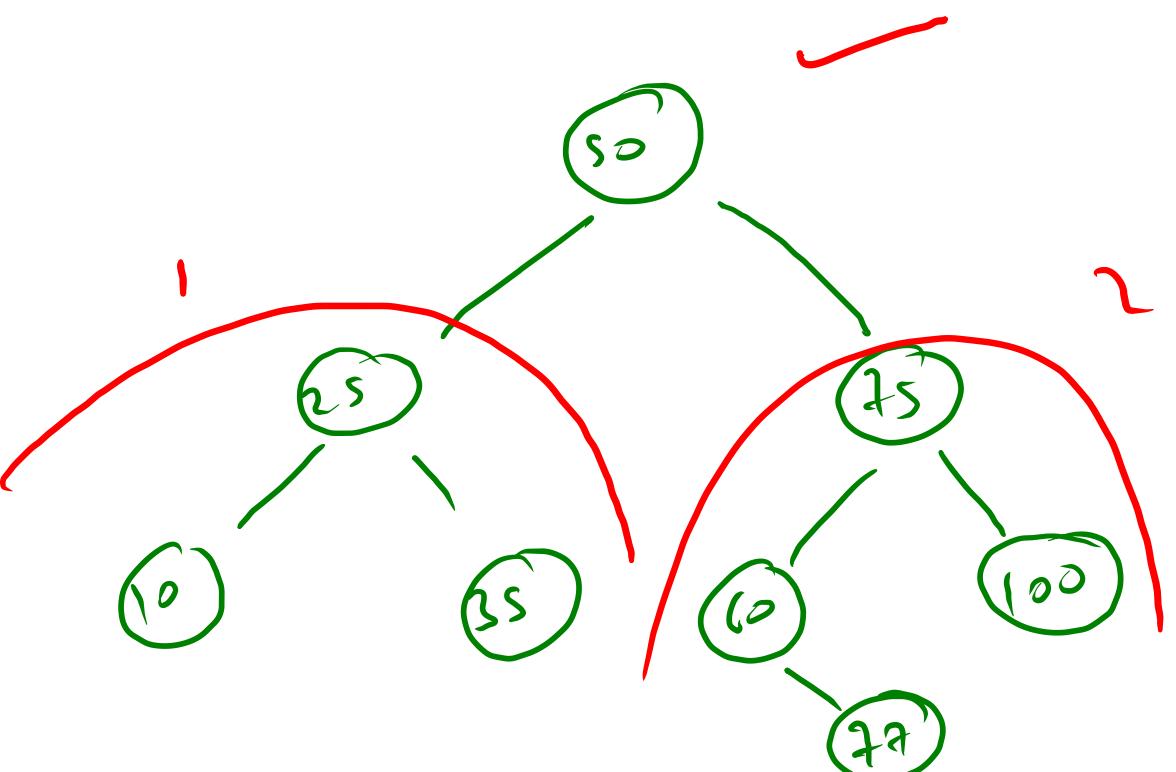
static Pair2 isBST2(Node node){
    if(node == null){
        return new Pair2(Integer.MAX_VALUE, Integer.MIN_VALUE, true);
    }

    Pair2 l = isBST2(node.left);
    if(l.max >= node.data || l.ok==false) return new Pair2(0,0, false);

    Pair2 r = isBST2(node.right);
    if(r.min <= node.data || r.ok==false) return new Pair2(0,0, false);
    F = 2 ✓
    return new Pair2(Math.min(l.min, node.data), Math.max(r.max, node.data), true);
}
∞   ∞

```

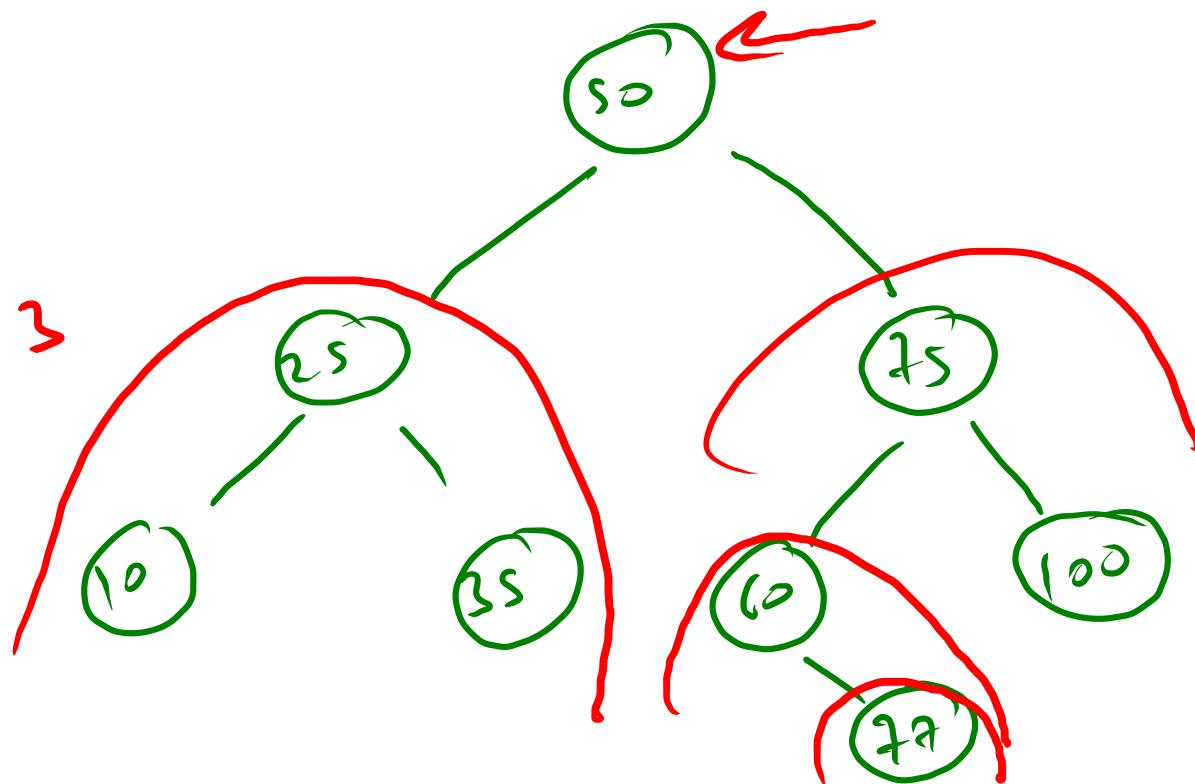




Ans

BSL

3



par2L

min

max

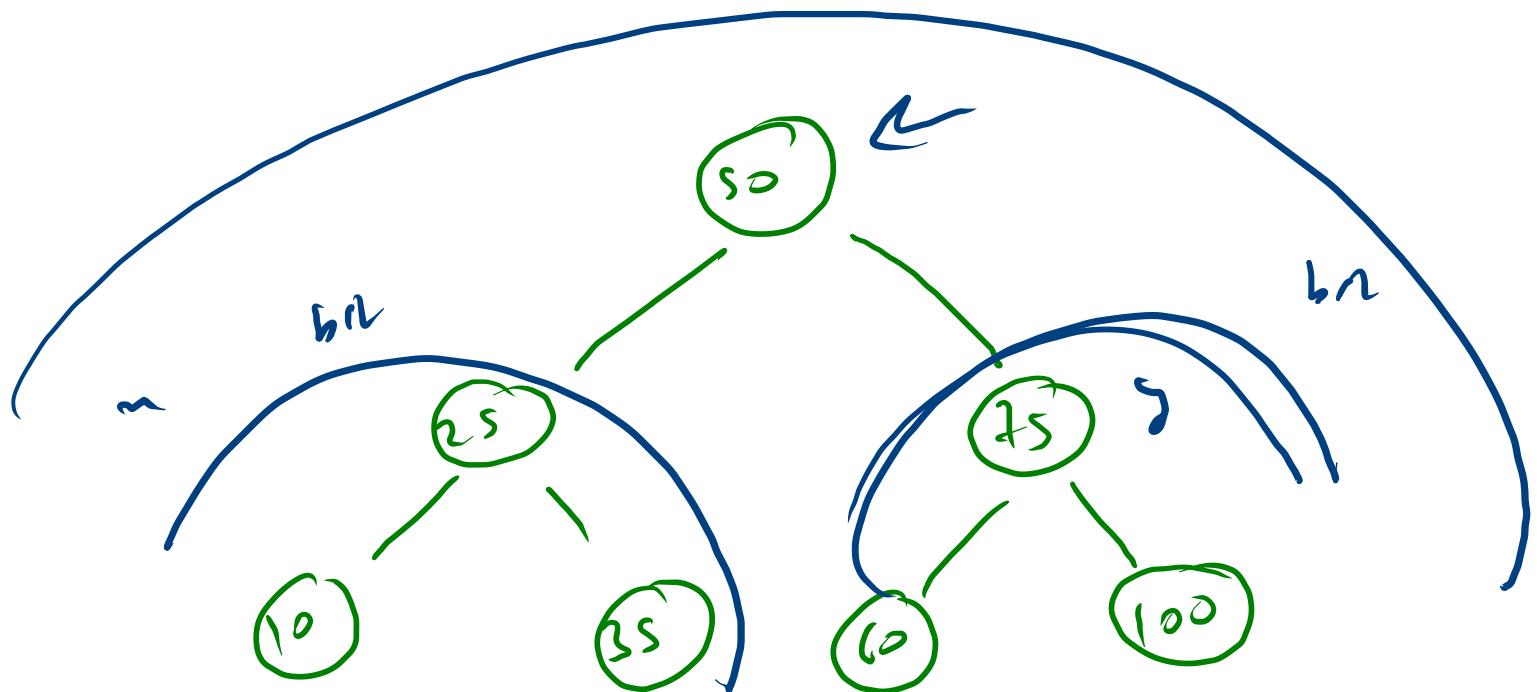
ishs2

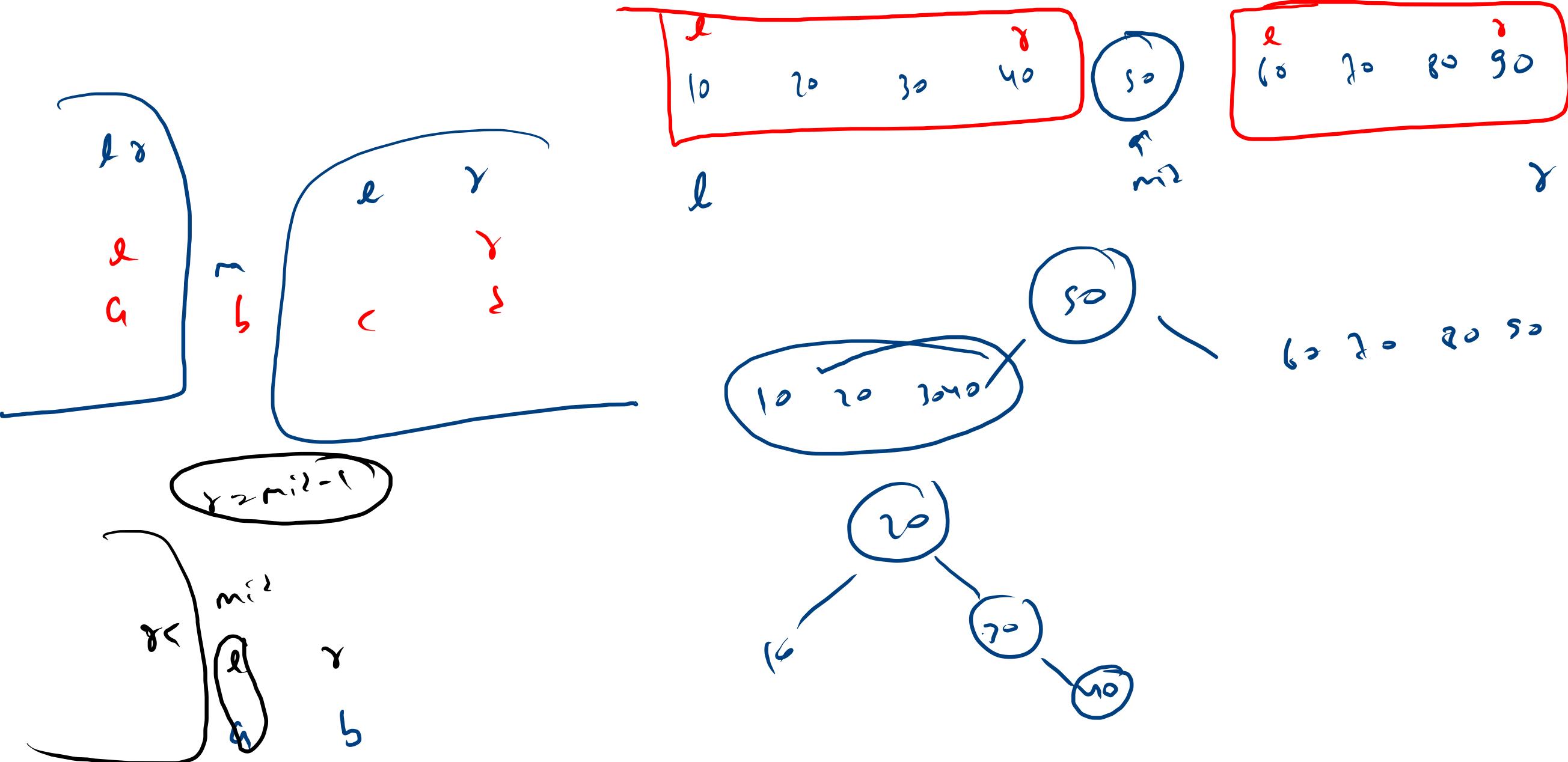
largestbit size

check

size

}





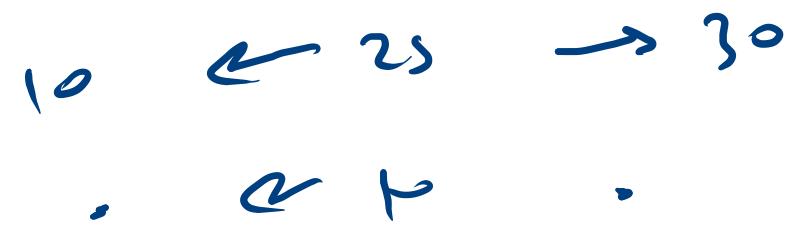
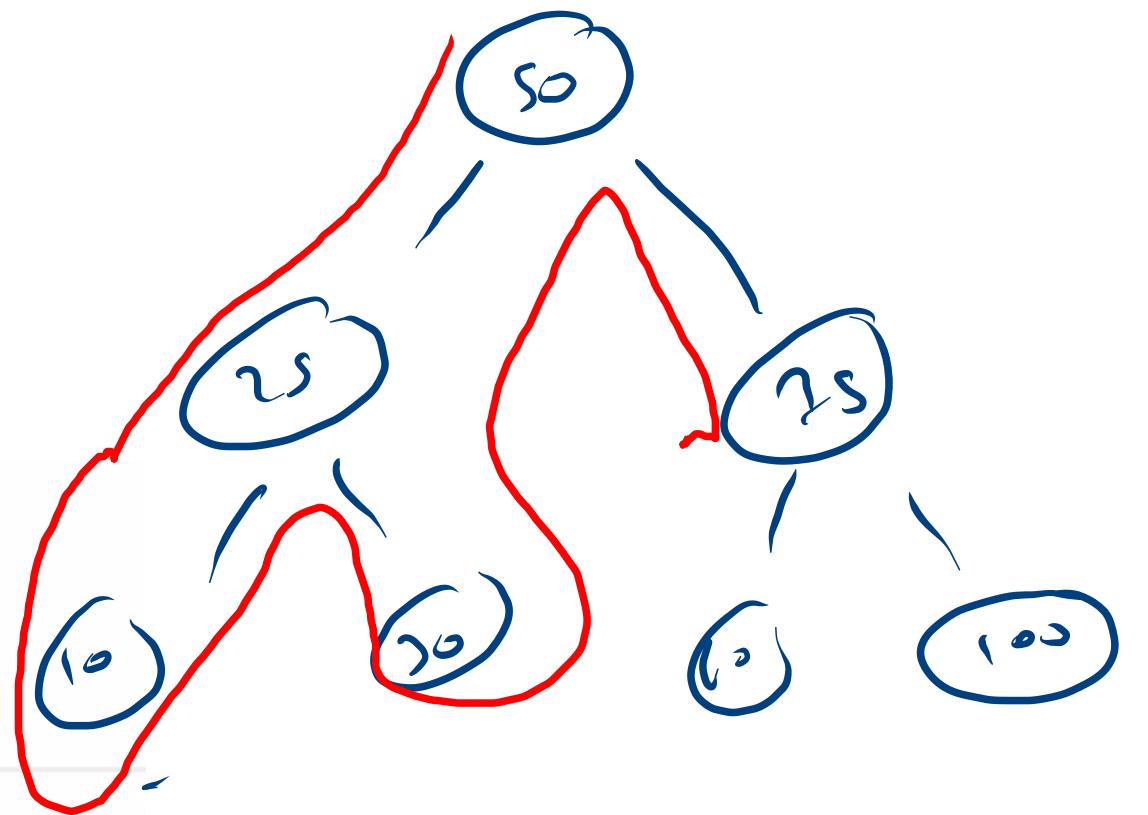
```

public static void display(Node node) {
    if (node == null) {
        return;
    }

    String str = "";
    str += node.left == null ? "." : node.left.data + "";
    str += " <- " + node.data + " -> ";
    str += node.right == null ? "." : node.right.data + "";
    System.out.println(str);

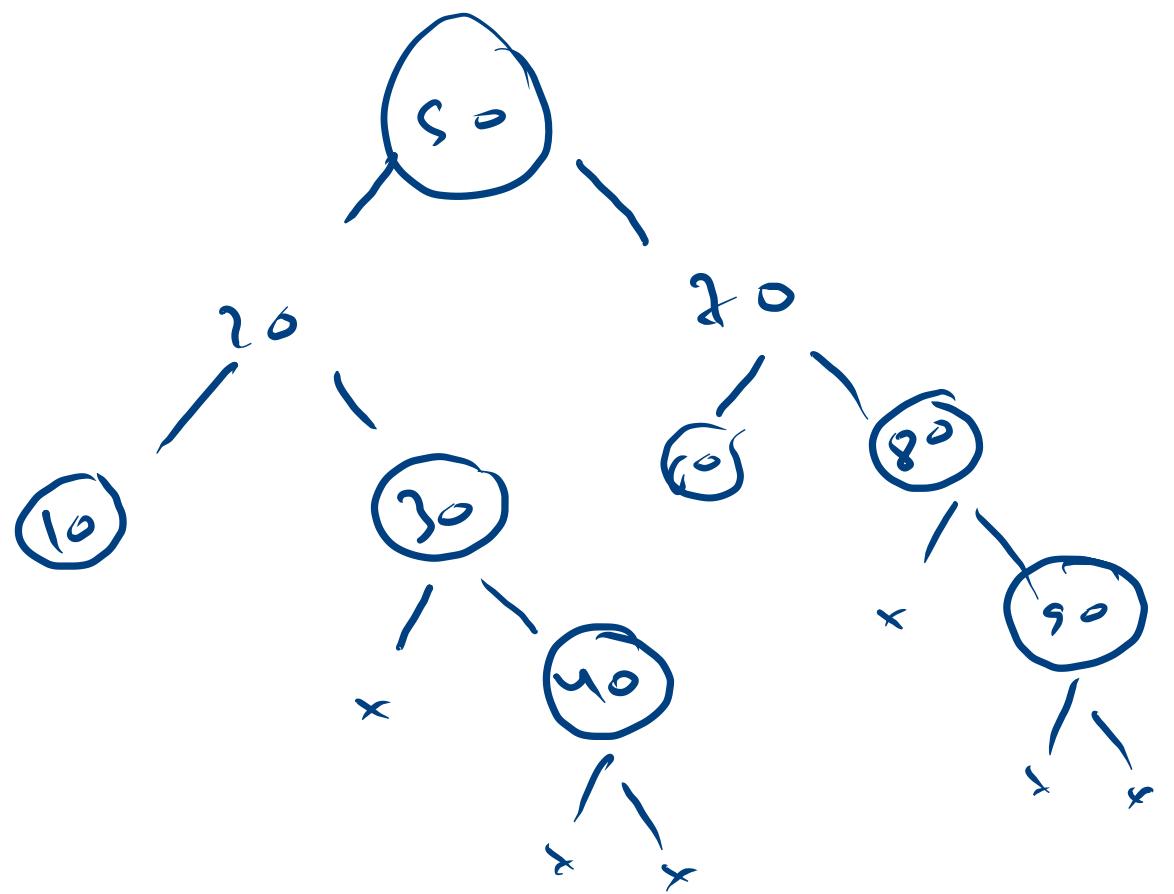
    display(node.left);
    display(node.right);
}

```

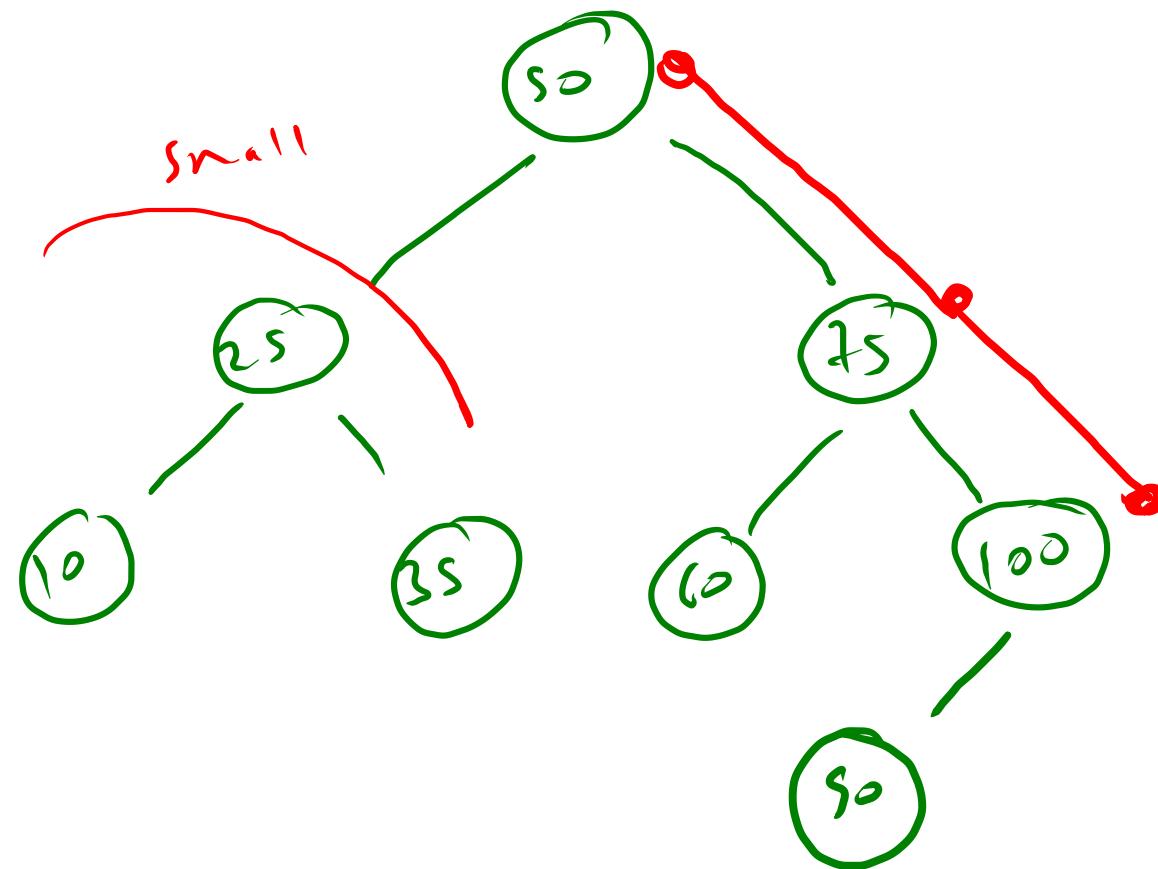


10 20 30 40 50 60 70 80 90

20 <- 50 -> 70
10 <- 20 -> 30
. <- 10 -> .
. <- 30 -> 40
. <- 40 -> .
60 <- 70 -> 80
. <- 60 -> .
. <- 80 -> 90
. <- 90 -> .

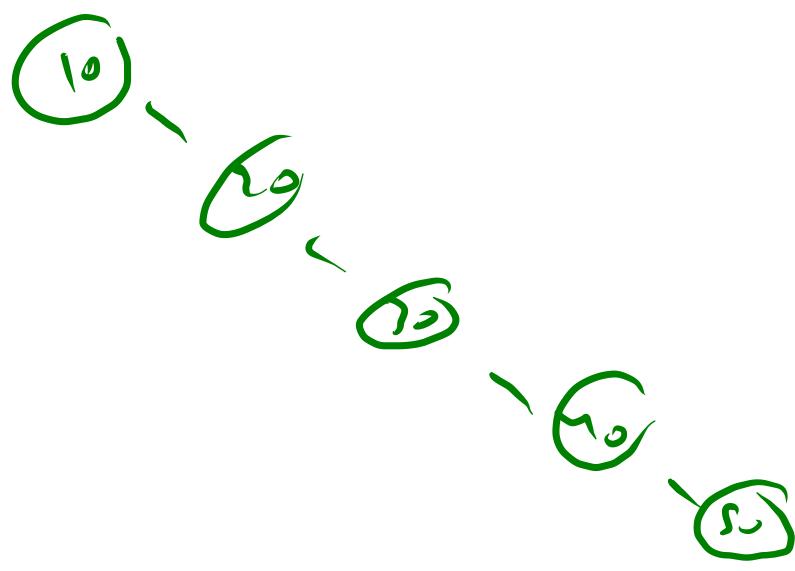
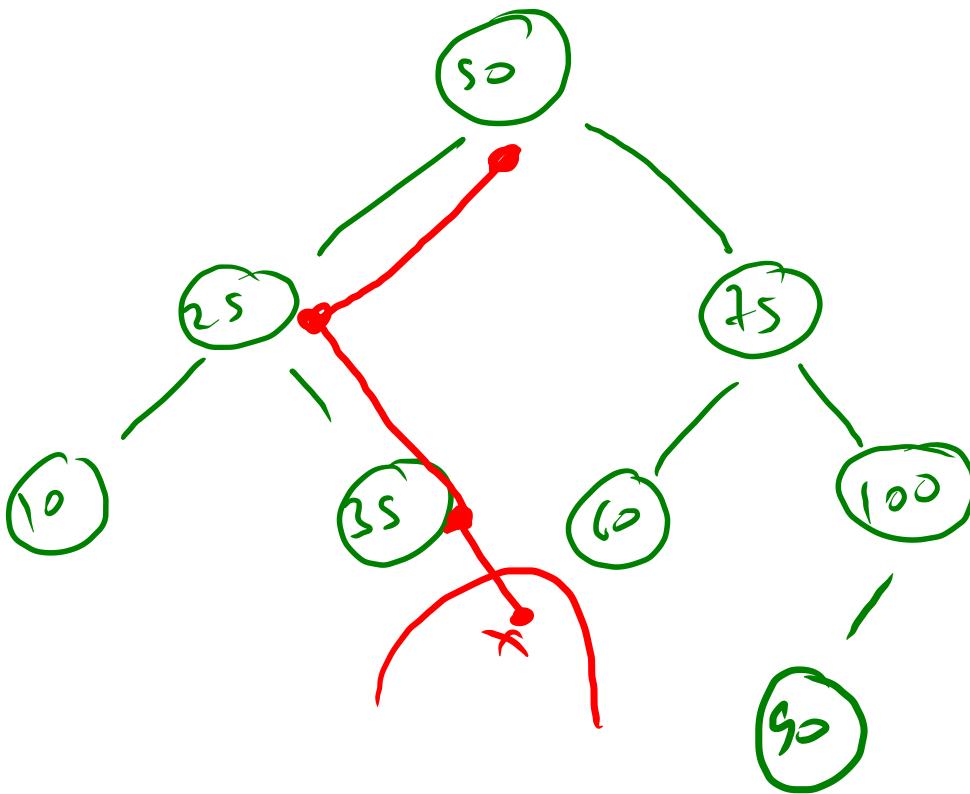


May



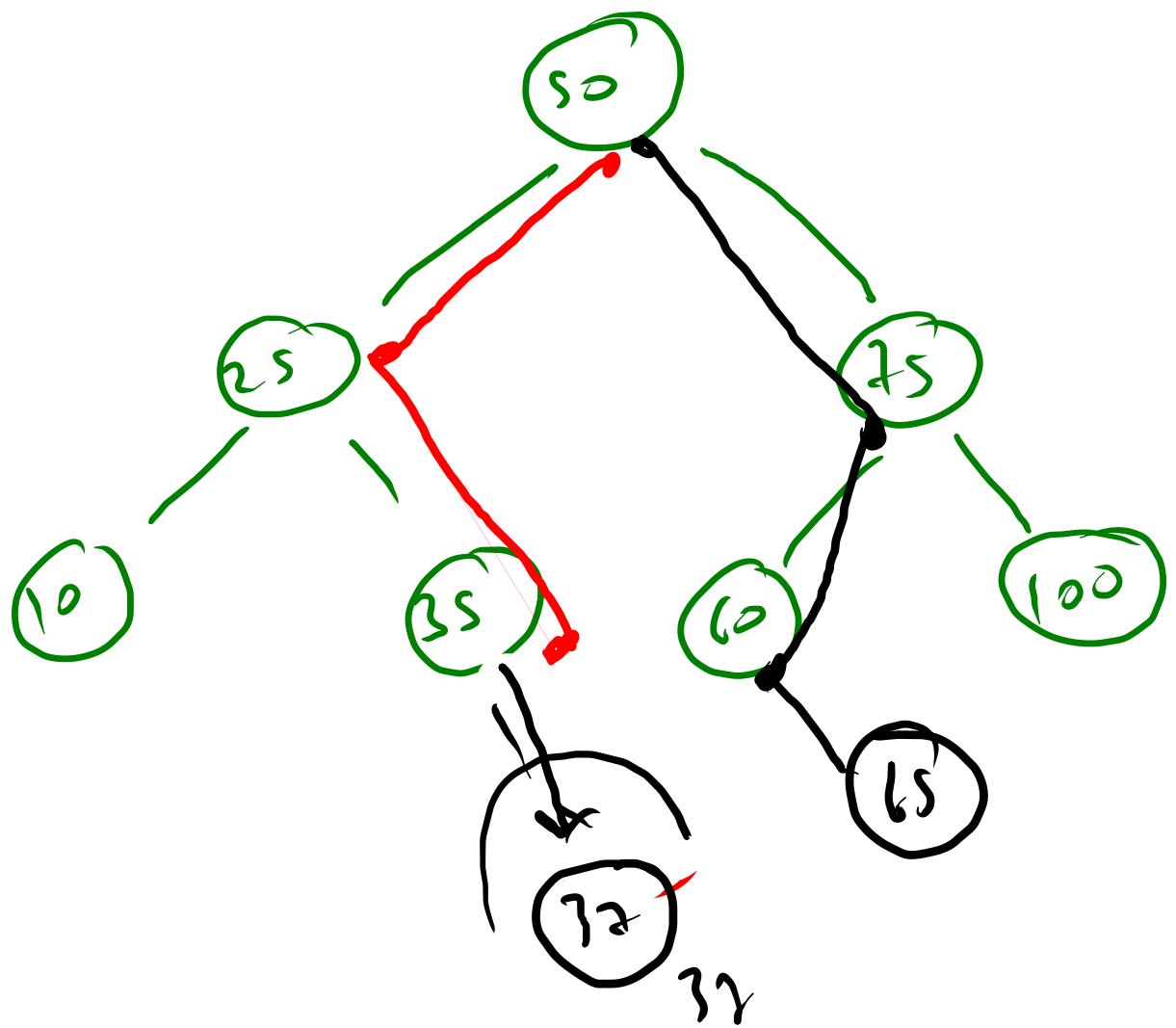
35
36

X

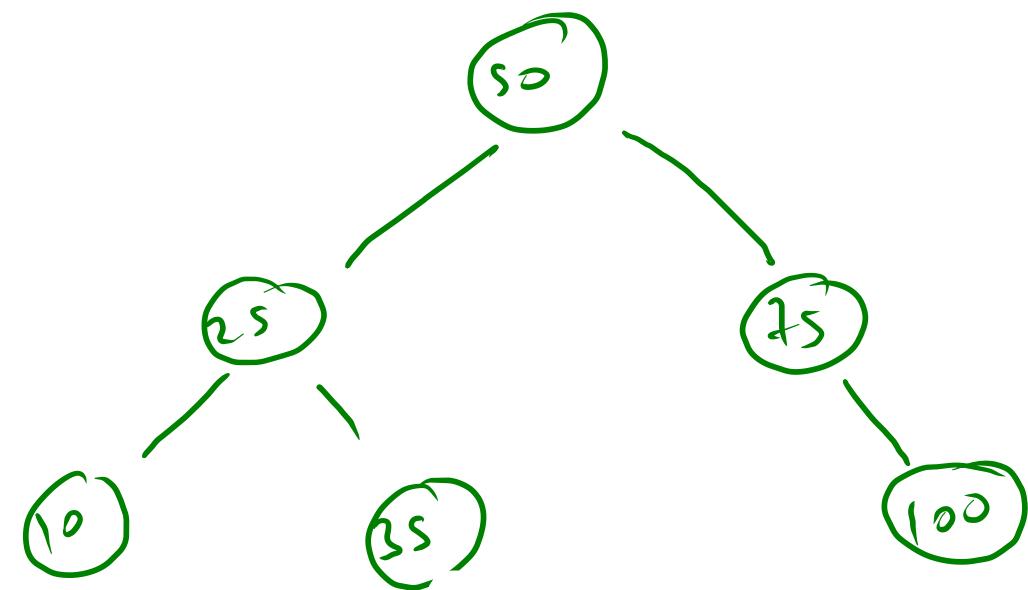
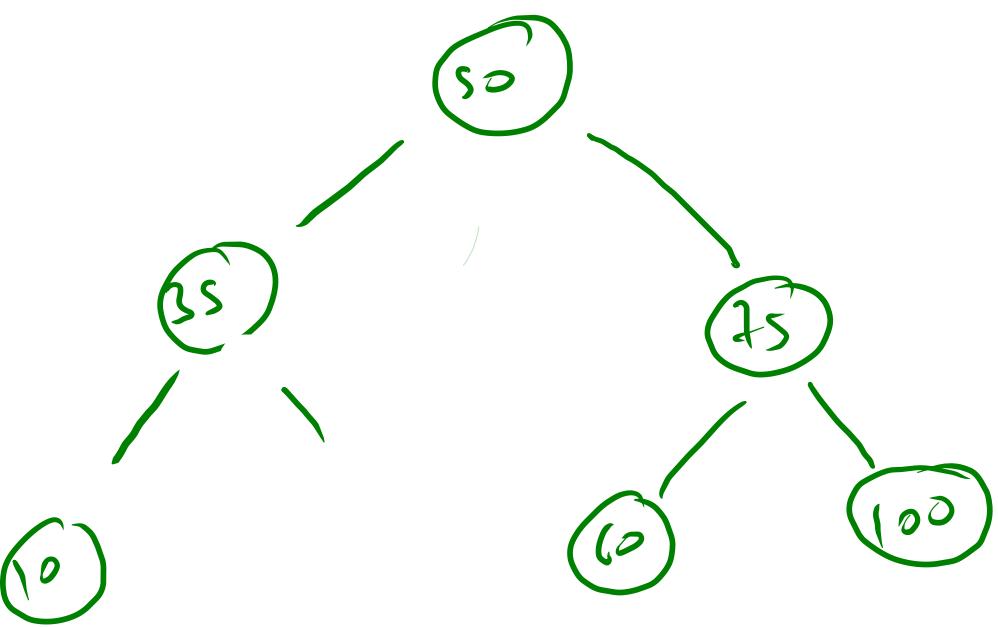


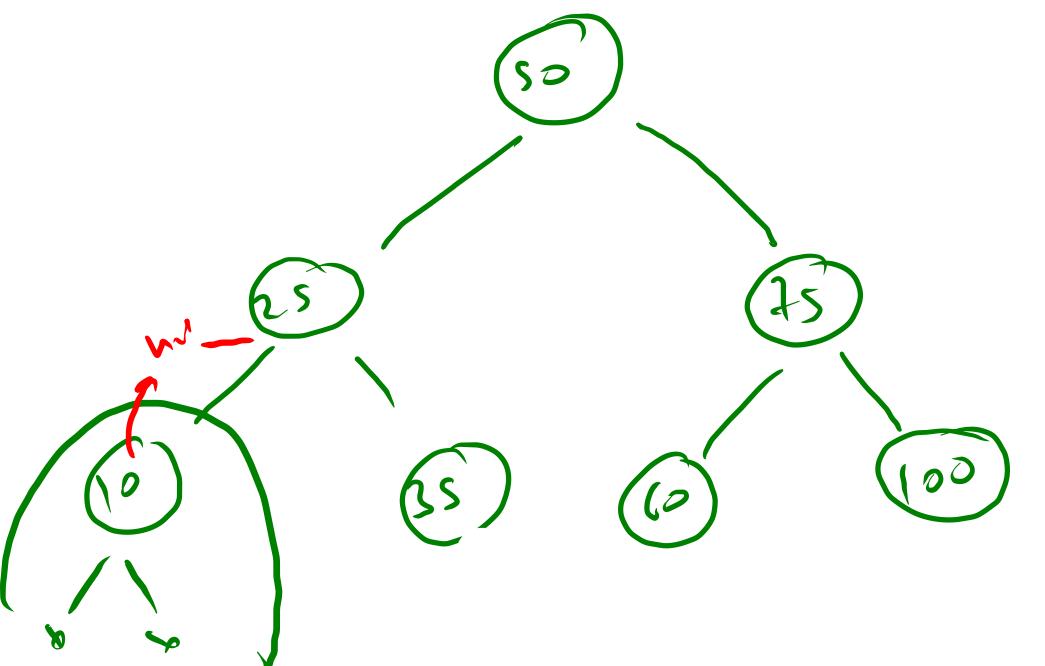
data = 32

data = 65



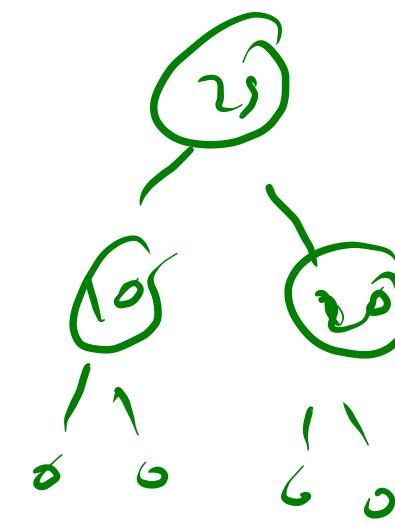
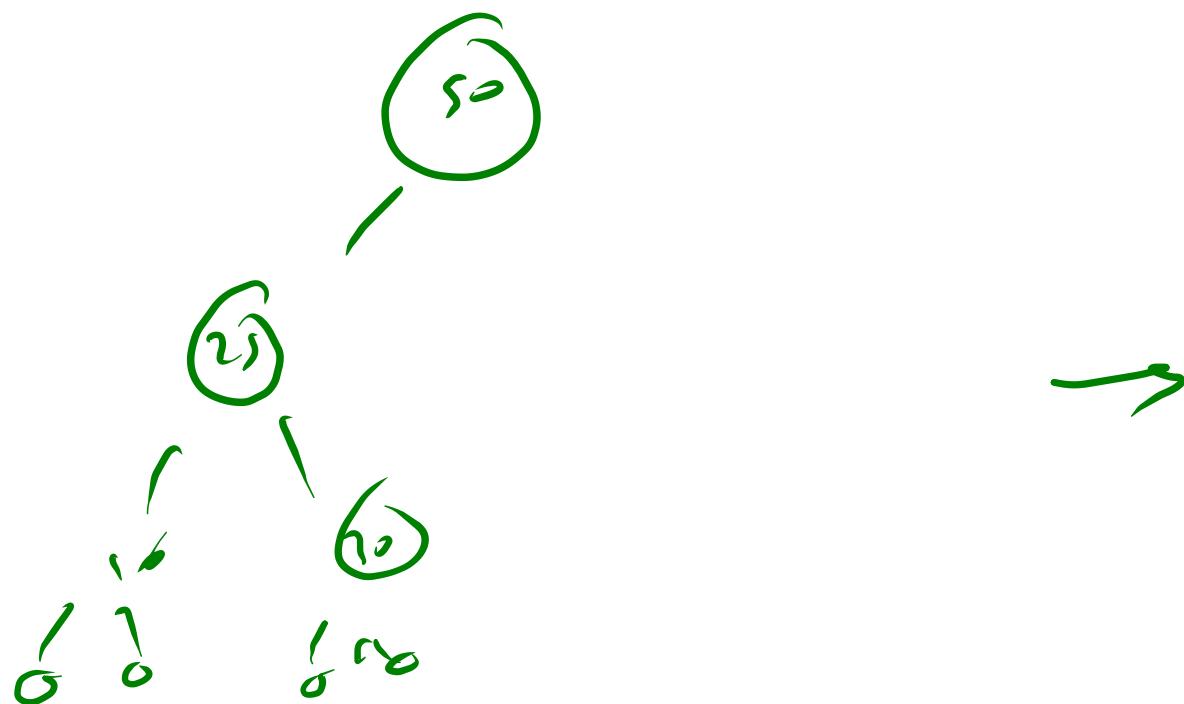
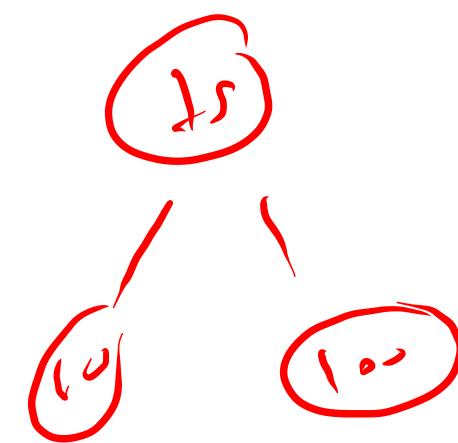
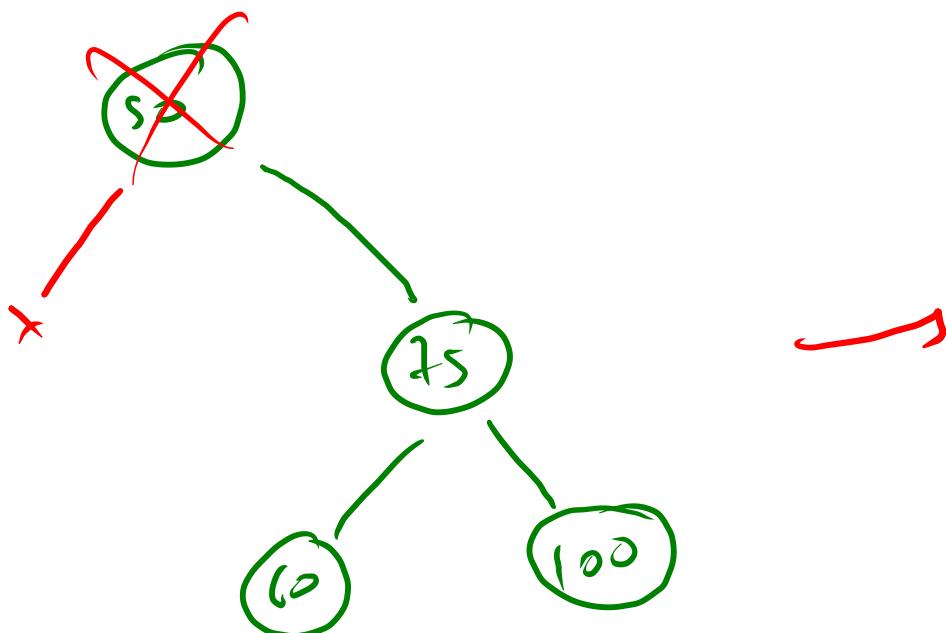
$d \text{ atm} =$ 60





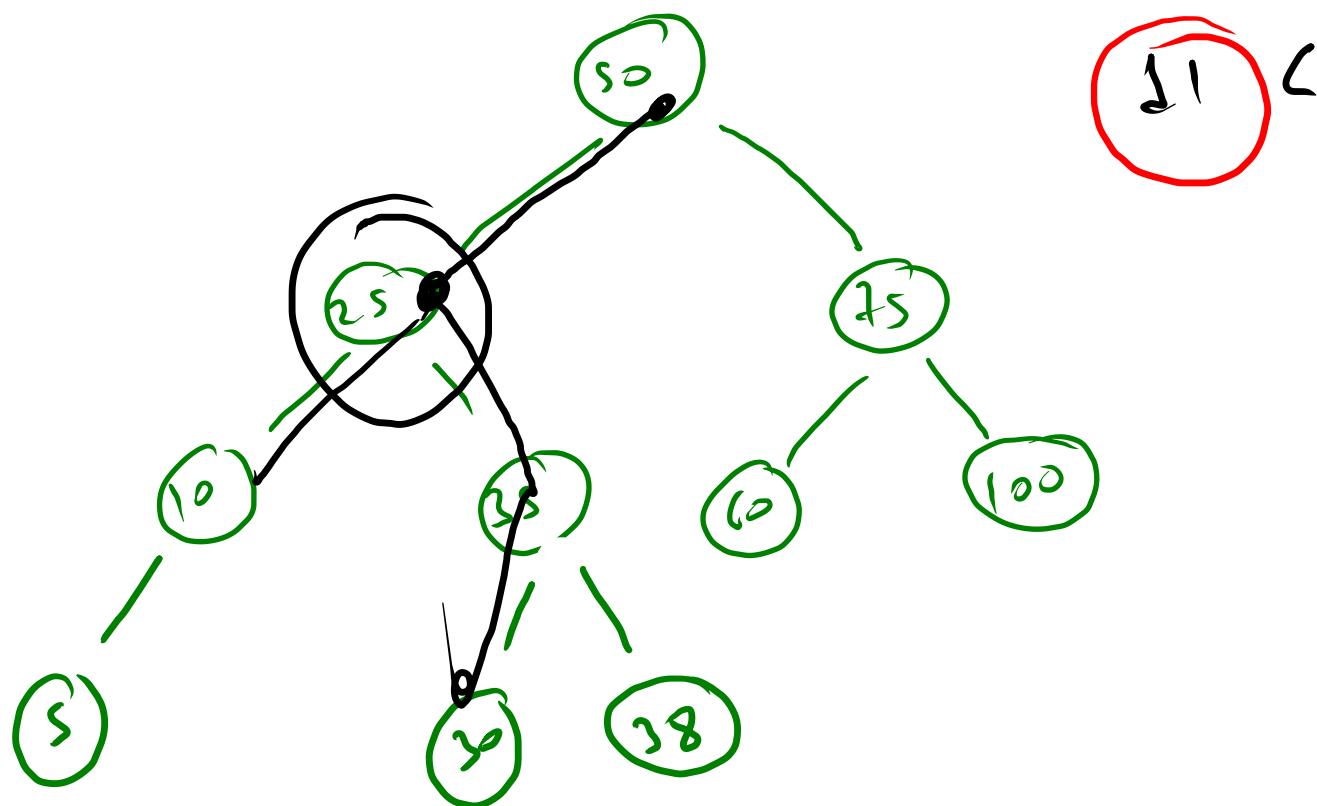
25 \rightarrow 10

~~10~~ \rightarrow *



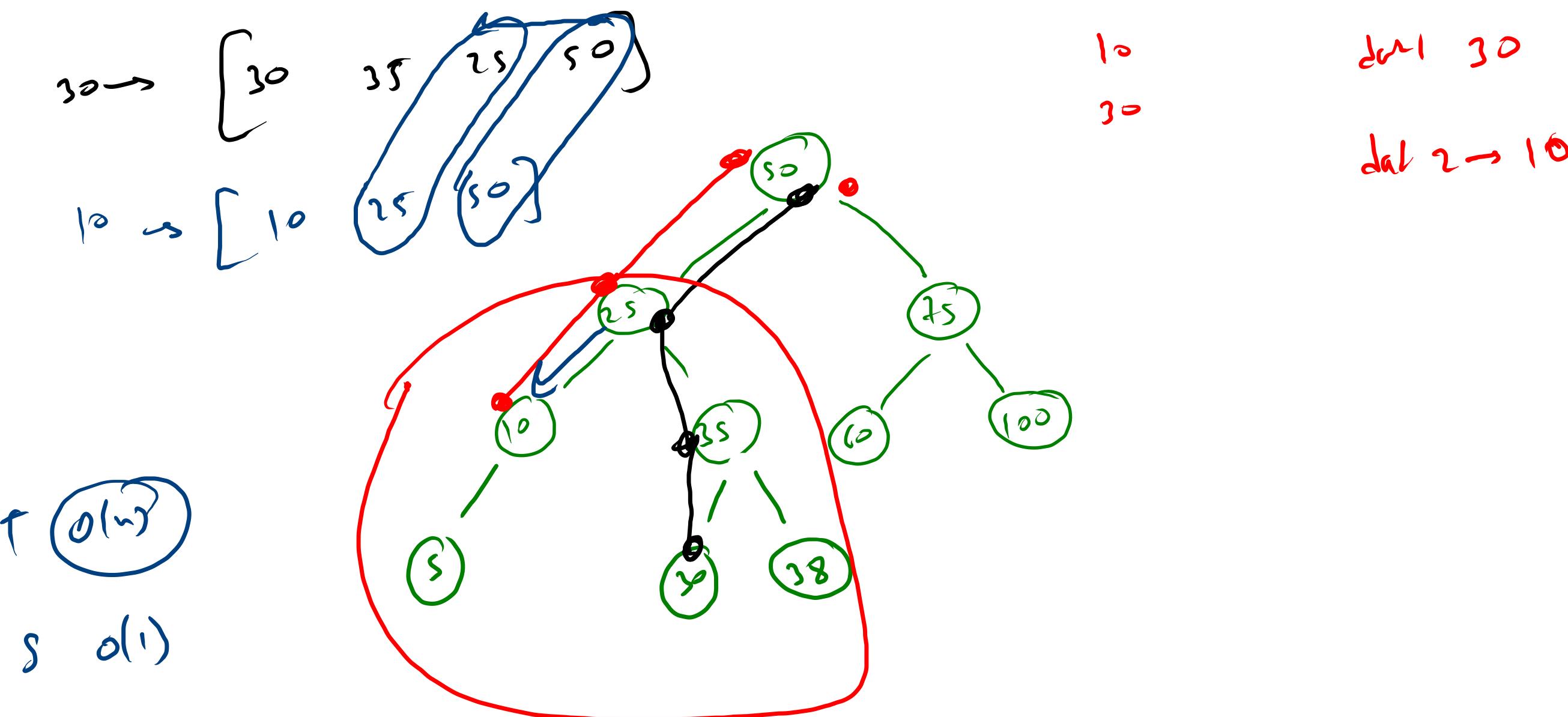
$d \rightarrow$
 $d_2 \rightarrow$

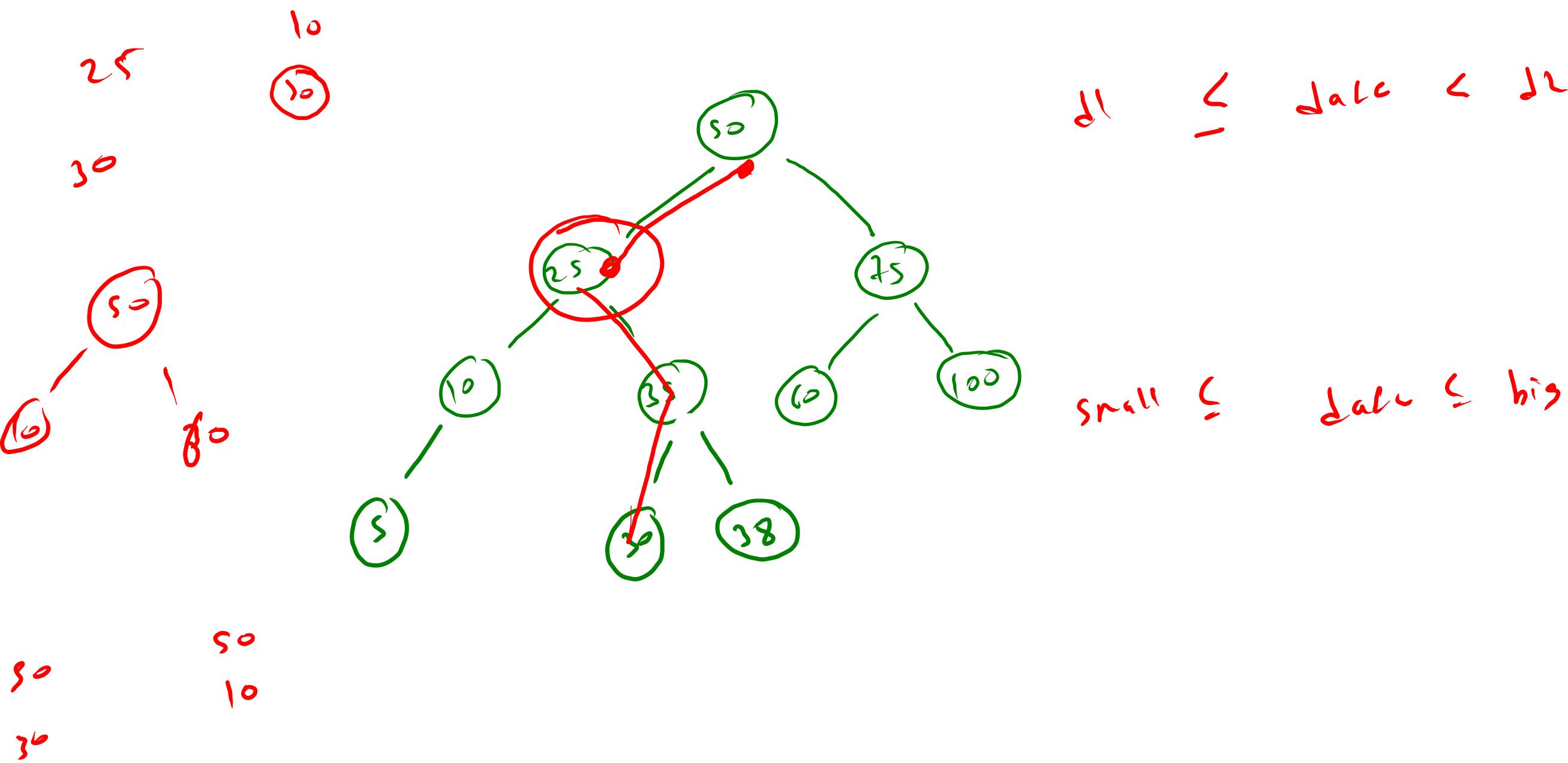
10
30



date

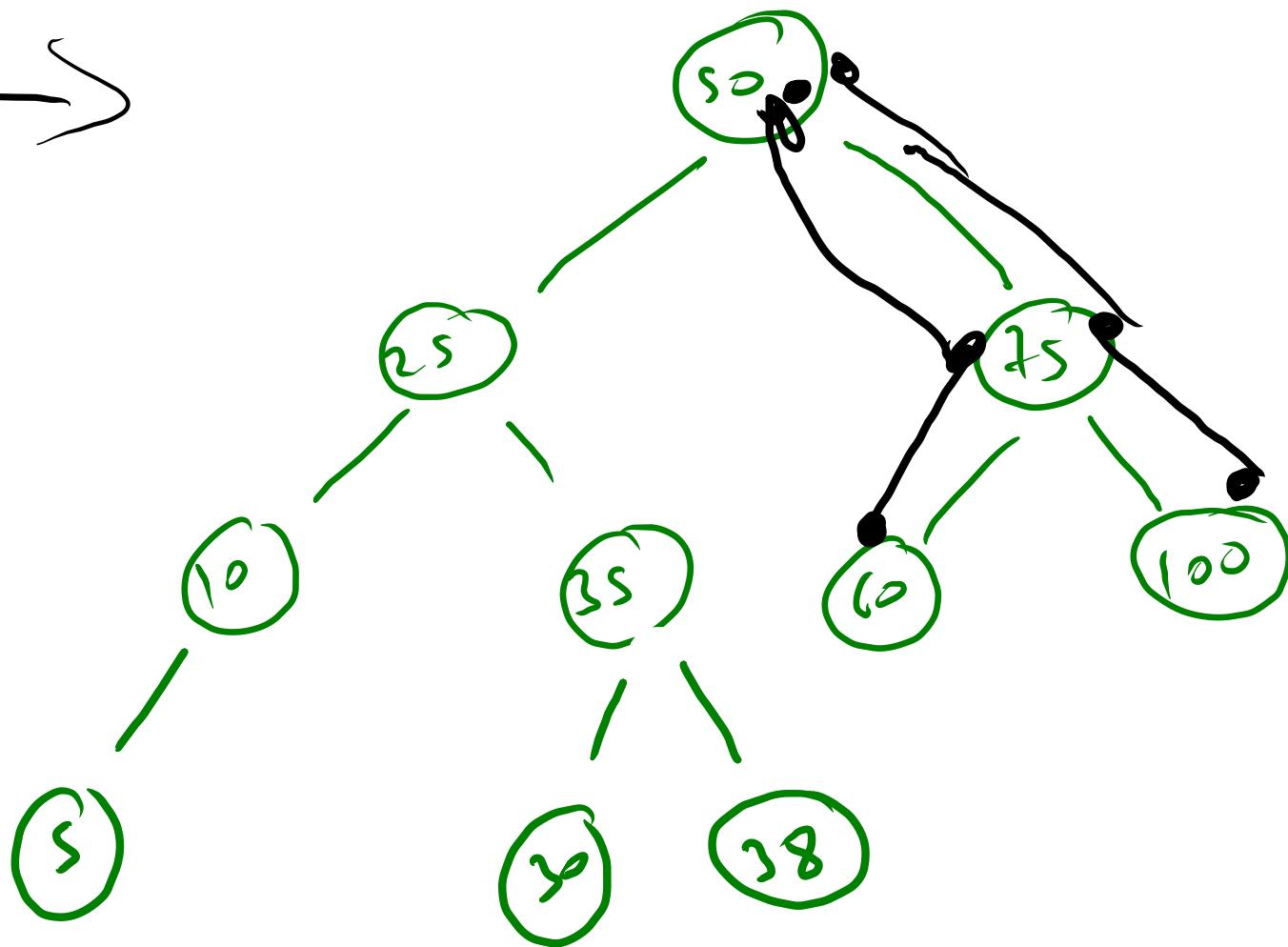
< 22





60

100

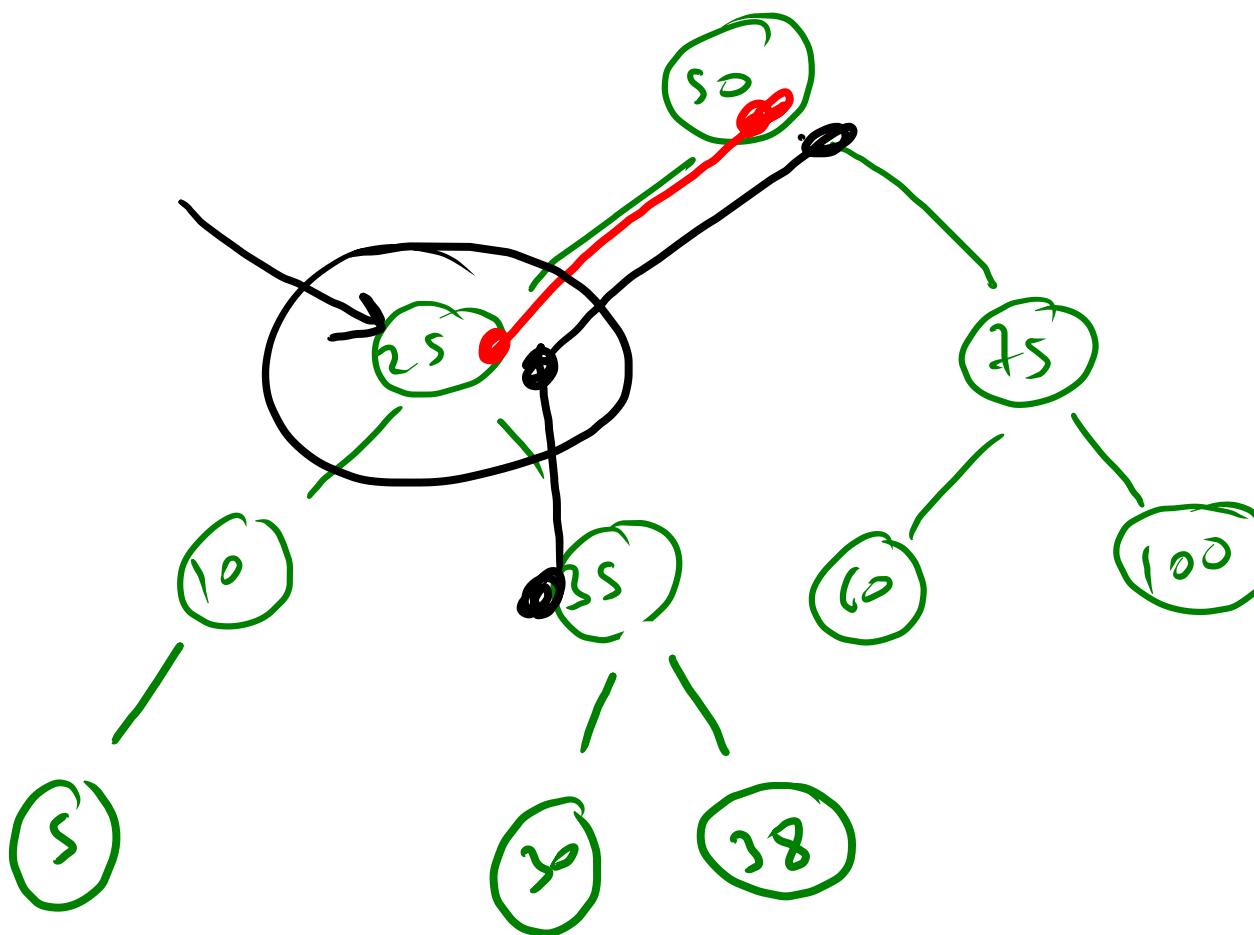


dul

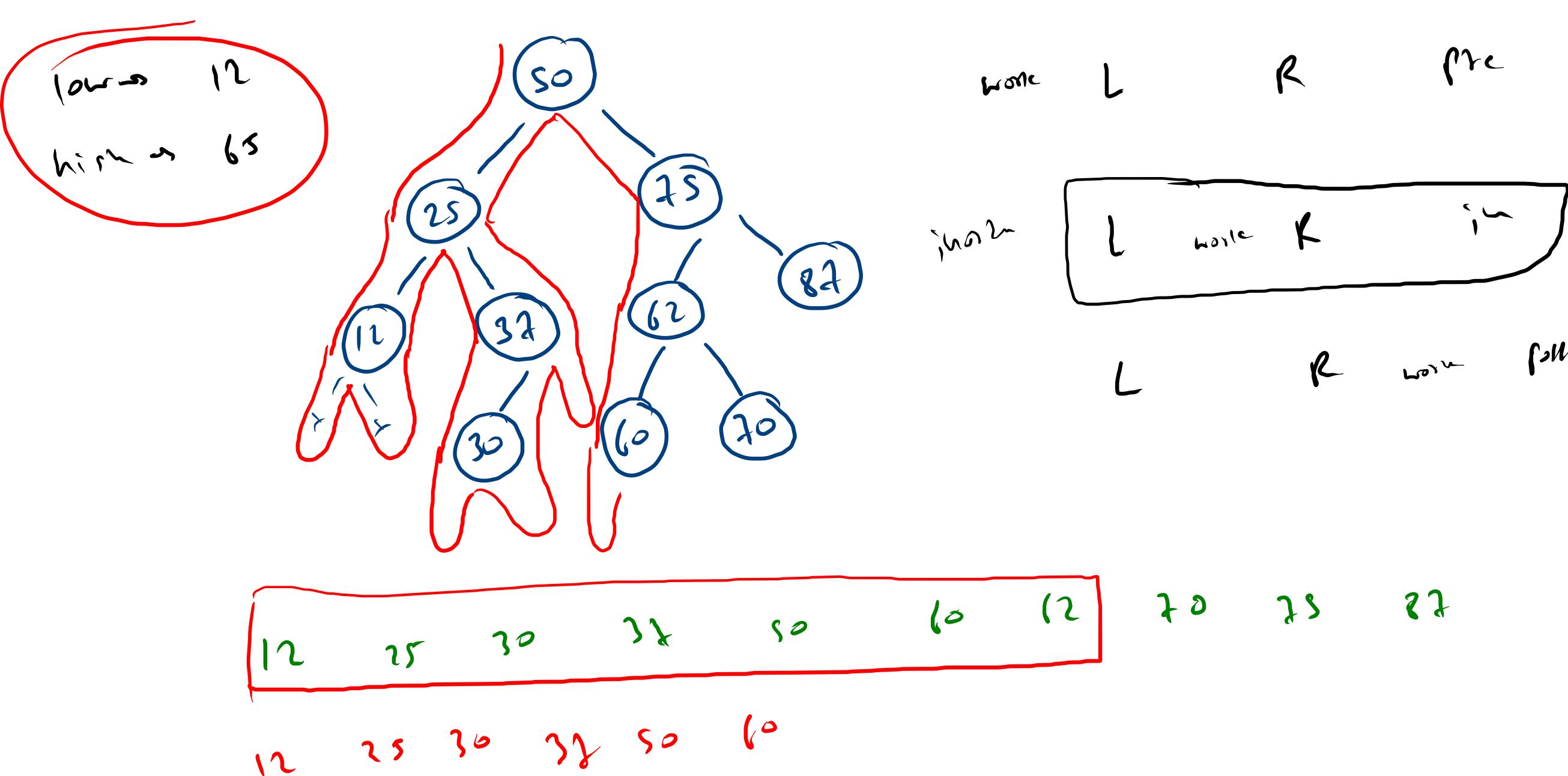
25

d2

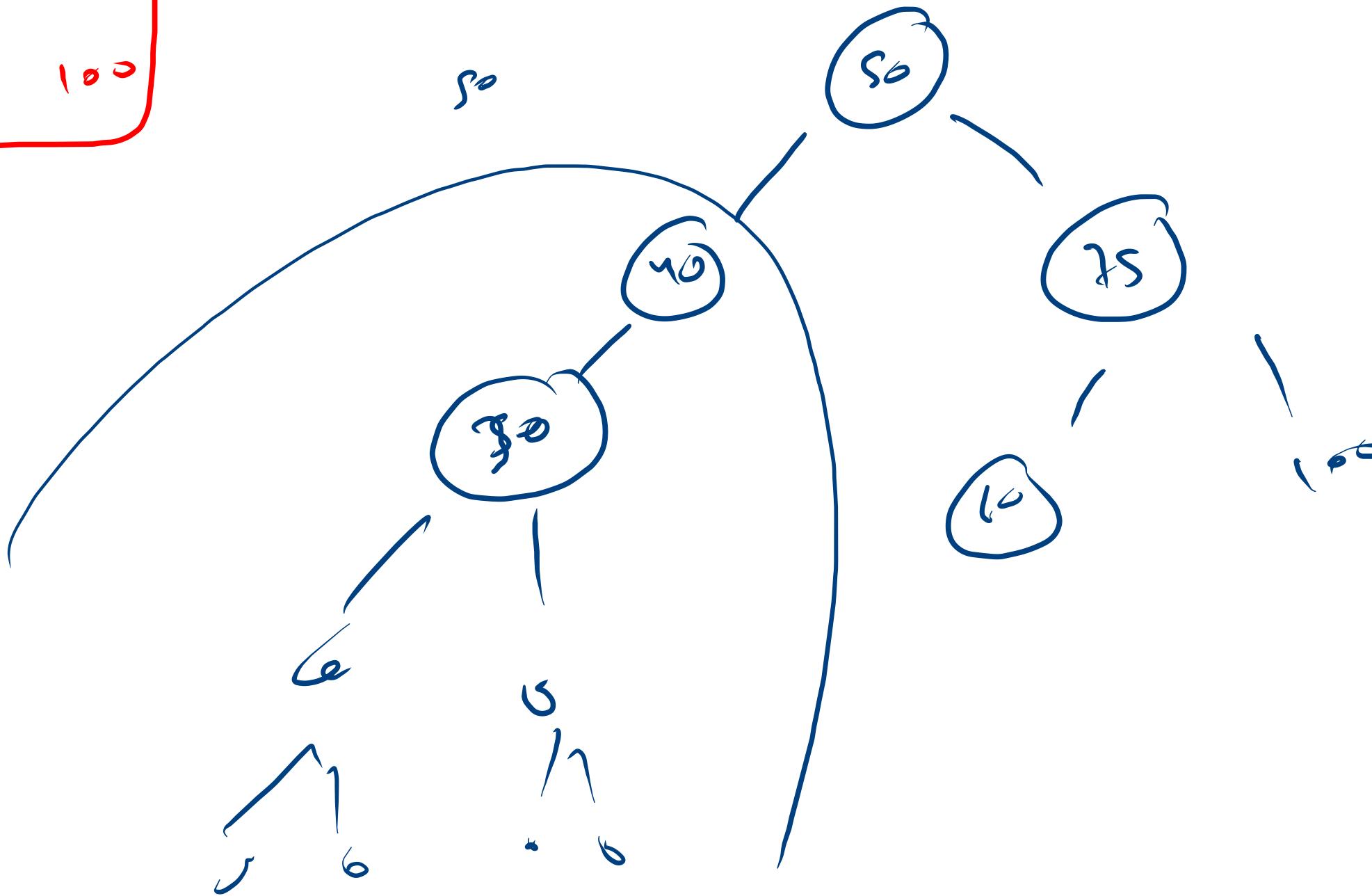
35



25 →



$$10 = 50$$
$$h^e = 100$$

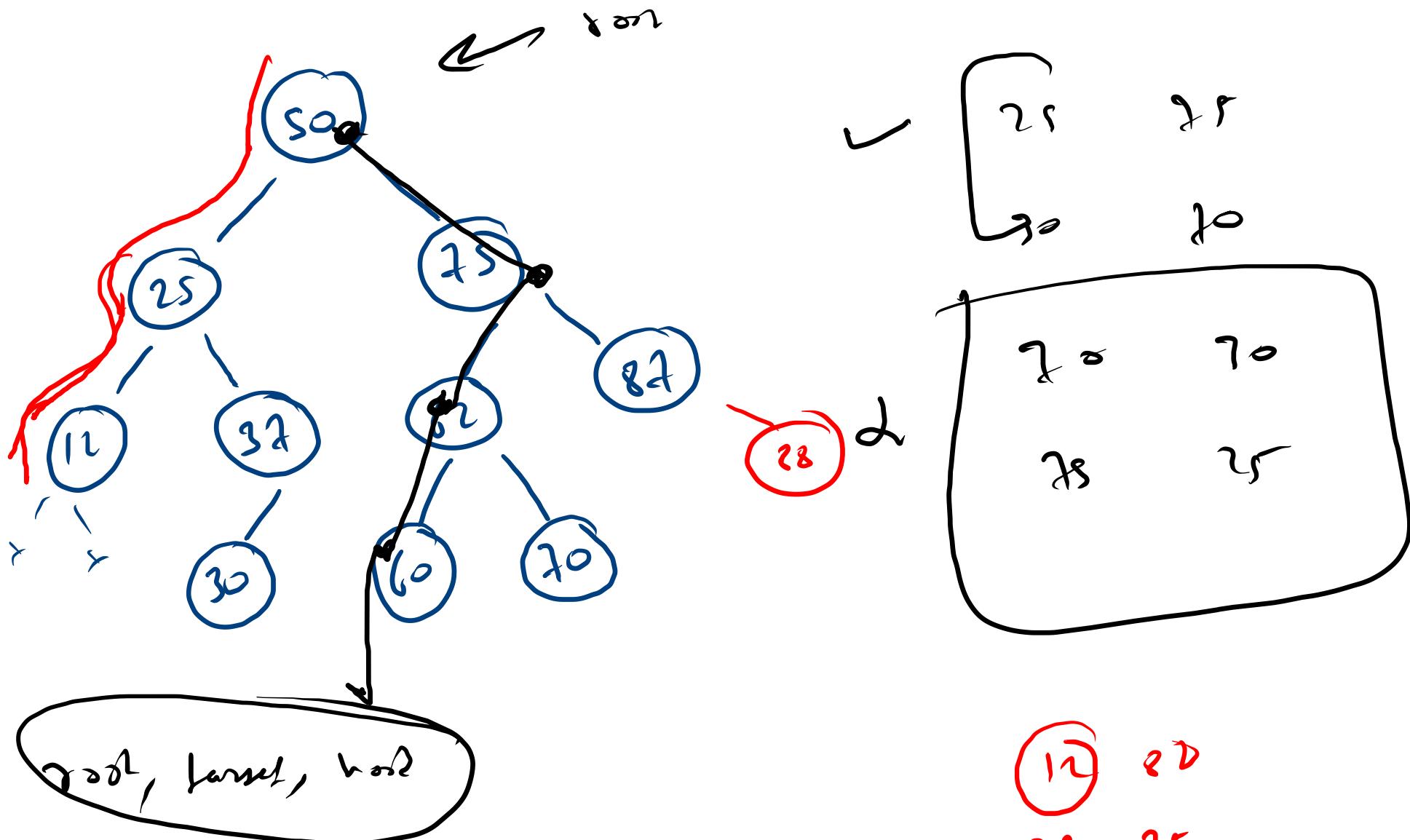


June 100

$$f_{4r} - 2r = 28$$

25 24
12 88

25



arrays

arrays list

0 ...

[0] =

int → *

"End" → 150

US

20

V/C

30

Bahs

15

HashMap

* → *

shifts

Integer

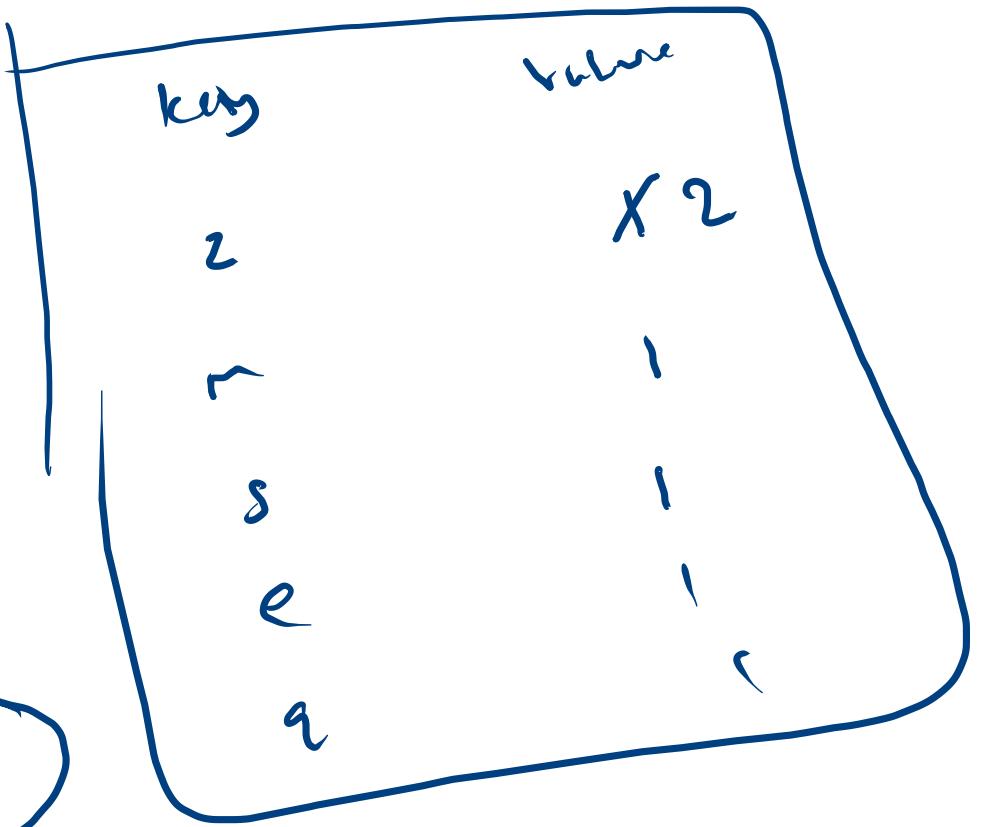
put (key, value)

get (key) $O(1)$

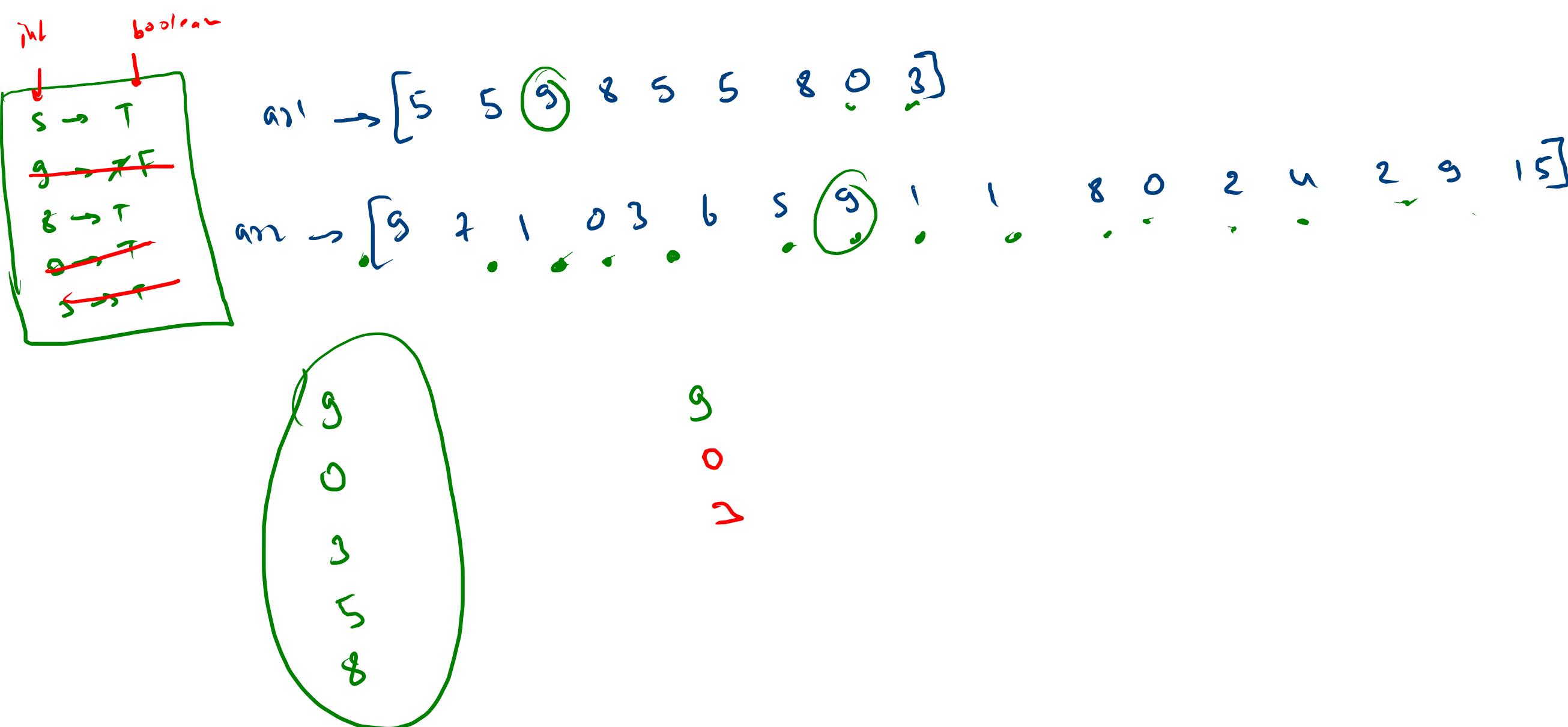
size ()

zmszedqxlzv
a - - 2

hashmap



containsKey



$1 \rightarrow x_{10}$
 $2 \rightarrow x_2$
 $3 \rightarrow 1$
 $5 \rightarrow r_0$
 $7 \rightarrow 1$

$a\pi \rightarrow [1, 1, 1, 2, 2, 2, 3, 5, 7]$
 $a\pi_2 \rightarrow [1, ?, !, !, 2, 2, 4]$

\downarrow
 s
 1
 2
 2

1...13

ar-[
12 ✓ s 1 2 ✓ 10 ✓ 2 ✓ 13 ✓ 7 ✓ 11 ✓ 8 ✓ 9 ✓ 11 ✓ 8 ✓ 9 ✓ 5 ✓ 6 ✓ 11 ✓]

1 2

5 6 7 8 9 10 11 12 13

0 2
0 → 2

5 → 6

0 11 12
1

4 5 6

1 2 3 ~ 100

200 ... 300

4
5
6

1 → 100

8

~~Priorities~~
~~decrease~~

remove $\rightarrow \log(n)$

get $\rightarrow \log(n)$

pick $\rightarrow O(1)$

size ()



remove $\rightarrow 10$

$\rightarrow 20$

$\rightarrow 5$

$\rightarrow 20$

n

[12 62 22 18 32 55 ✓ 11 77 58 62 31 84 99]

13 12 62 22 15
37 99 11 37 98
67 31 84 99 4

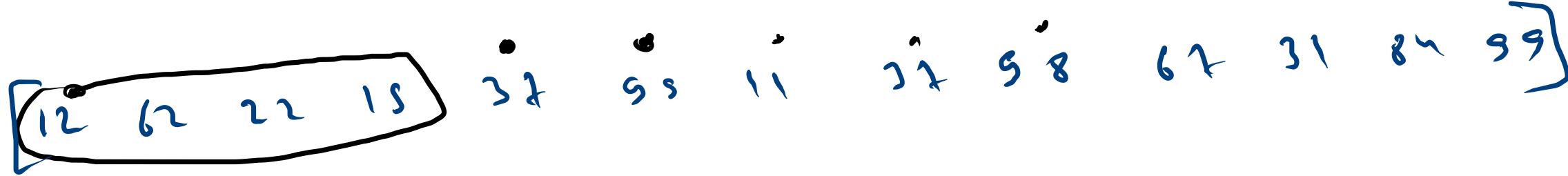
k--4

k for(n)

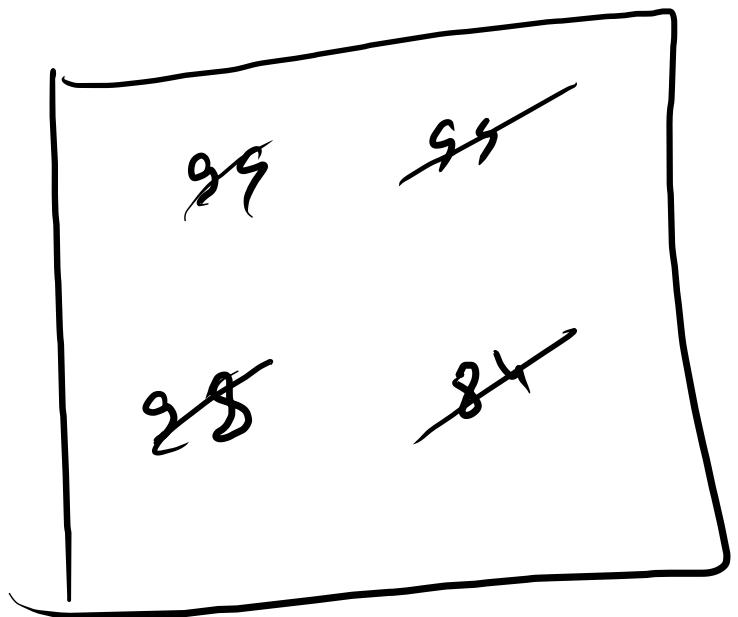
84
98
59
99

99
98
99
84

$k = n$

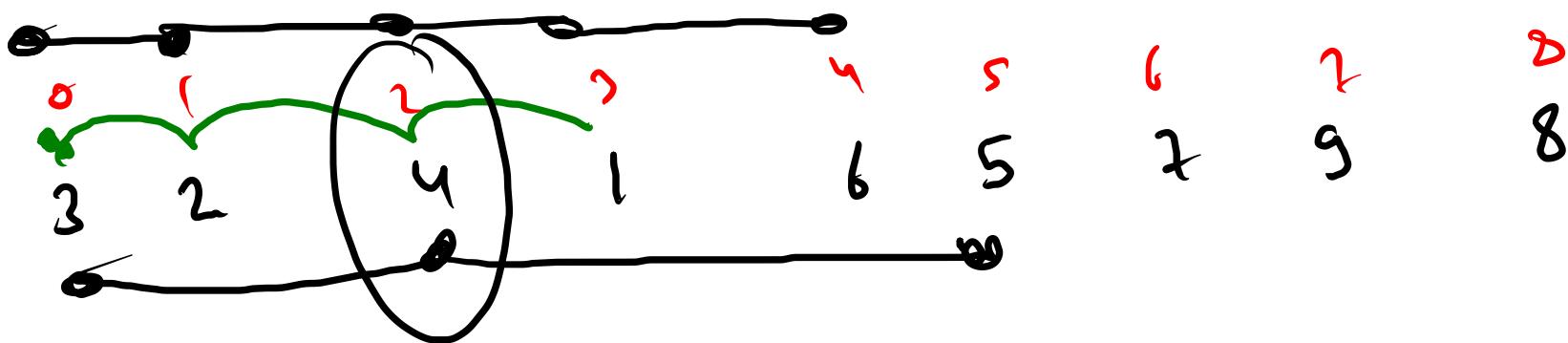


$k+1 \approx k$



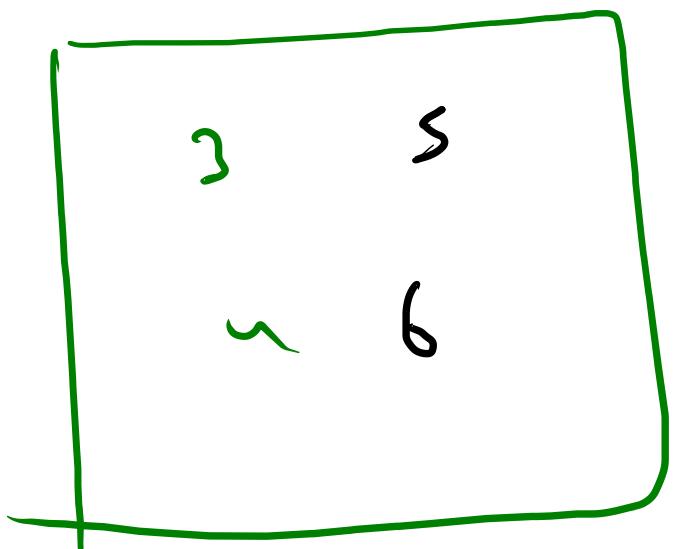
$n \log(k)$

82
92
72
22

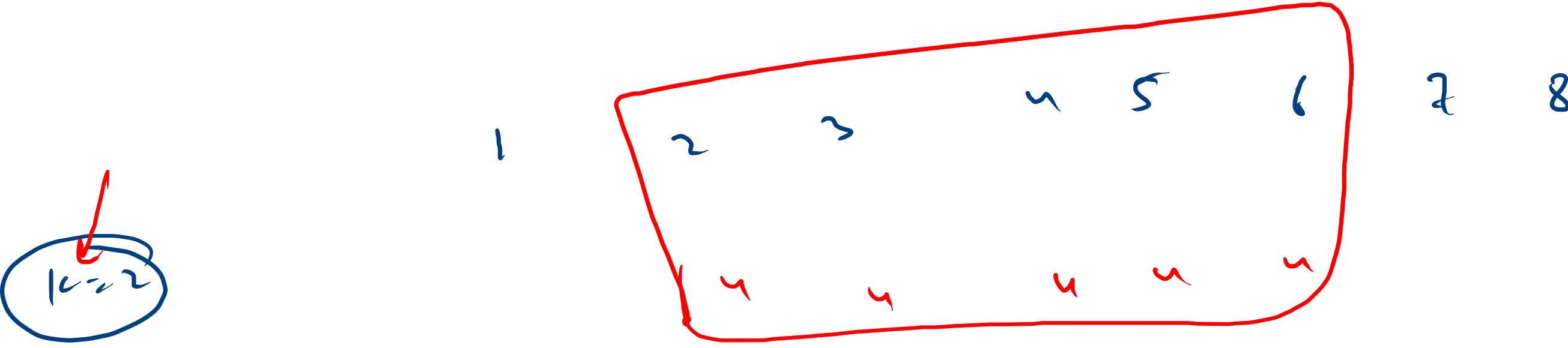


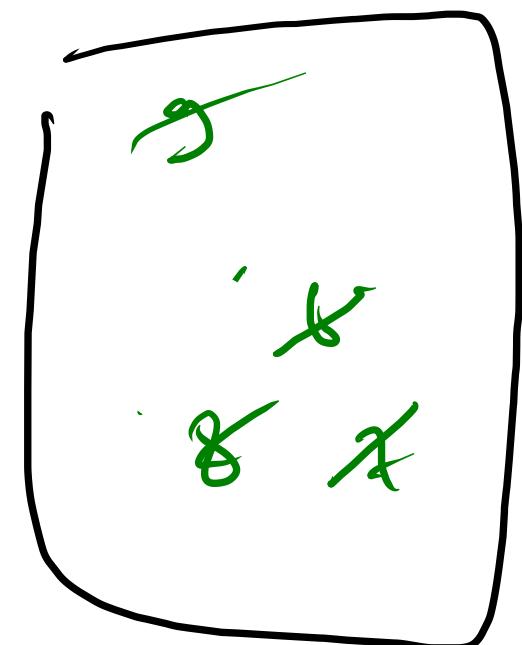
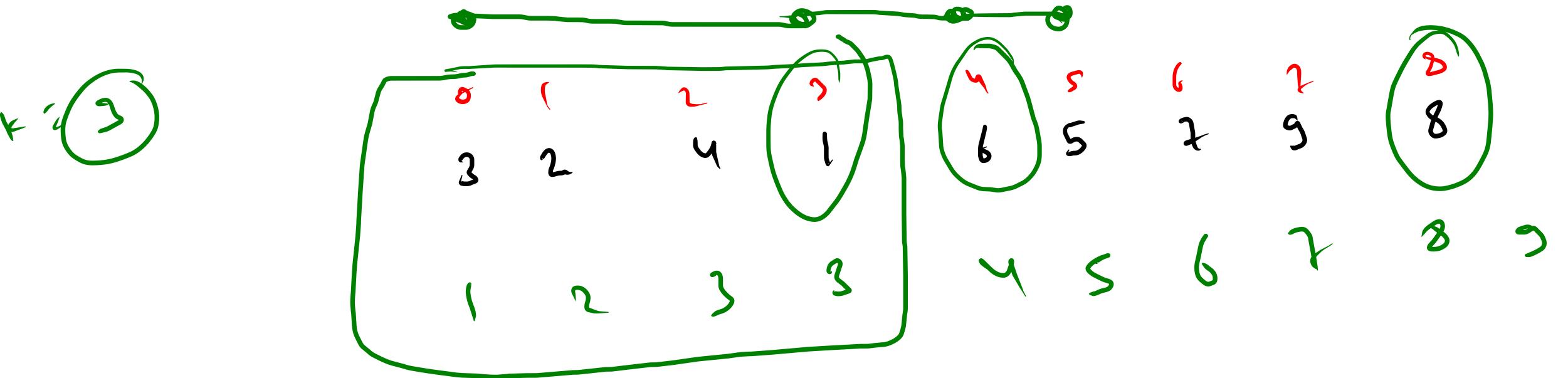
$k=3$

1 2



9 3 2 4 1 6 5 7 9
8 3





1c<1

y_k [10 20 30 40 50]

for value
10

s_k [5 7 9 11 19 55 57]

s_k 5

[1 2 3]

10
5
1

[1 2]

u_{lc} [8
10 20 30 40 50]

s_{lc} [5
7 9 11 19 55 52]

g_{lc} [i
j 2 3]

h_{lc} [32 39]

|

{ lk, 0 }

{ lk, 0 }

{ slc, 0 }

{ u_{lc}, 0 }

10 20

10 20 15 14

12

10 40

- ✓ add 10 add 20
- ✓ add 30 add 40
- peek add 50
- peek remove
- peek remove
- peek remove
- peek remove
- peek quit

20

30

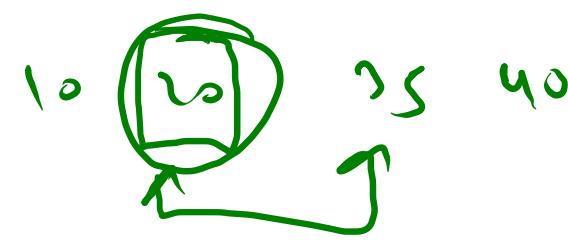
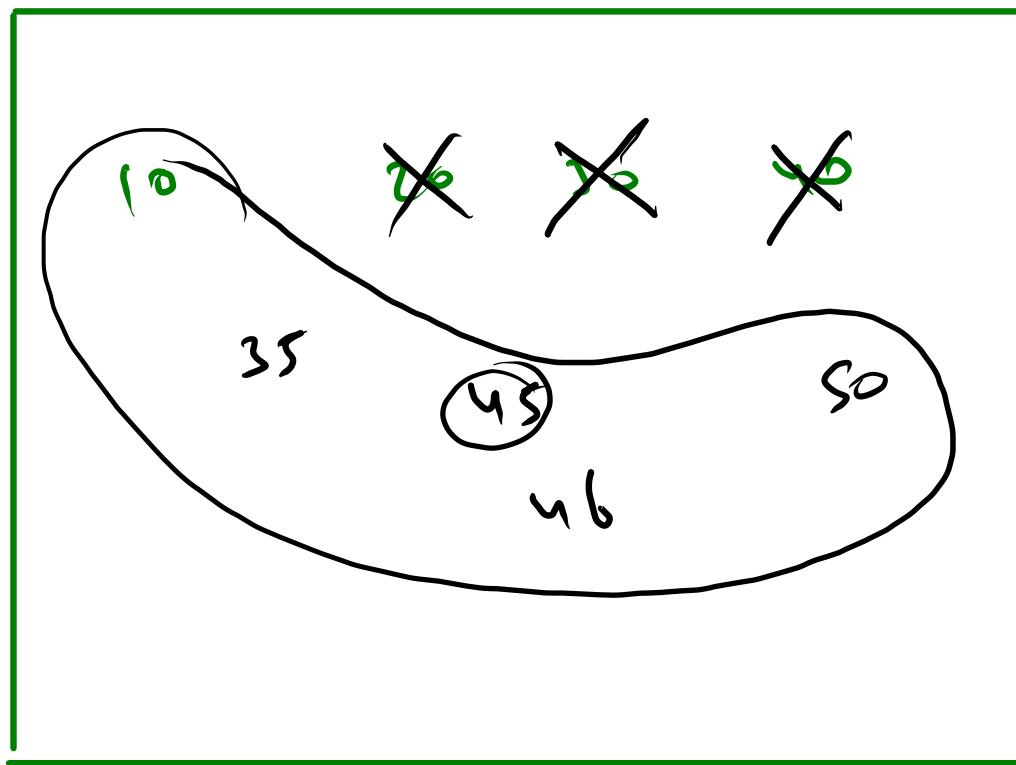
30

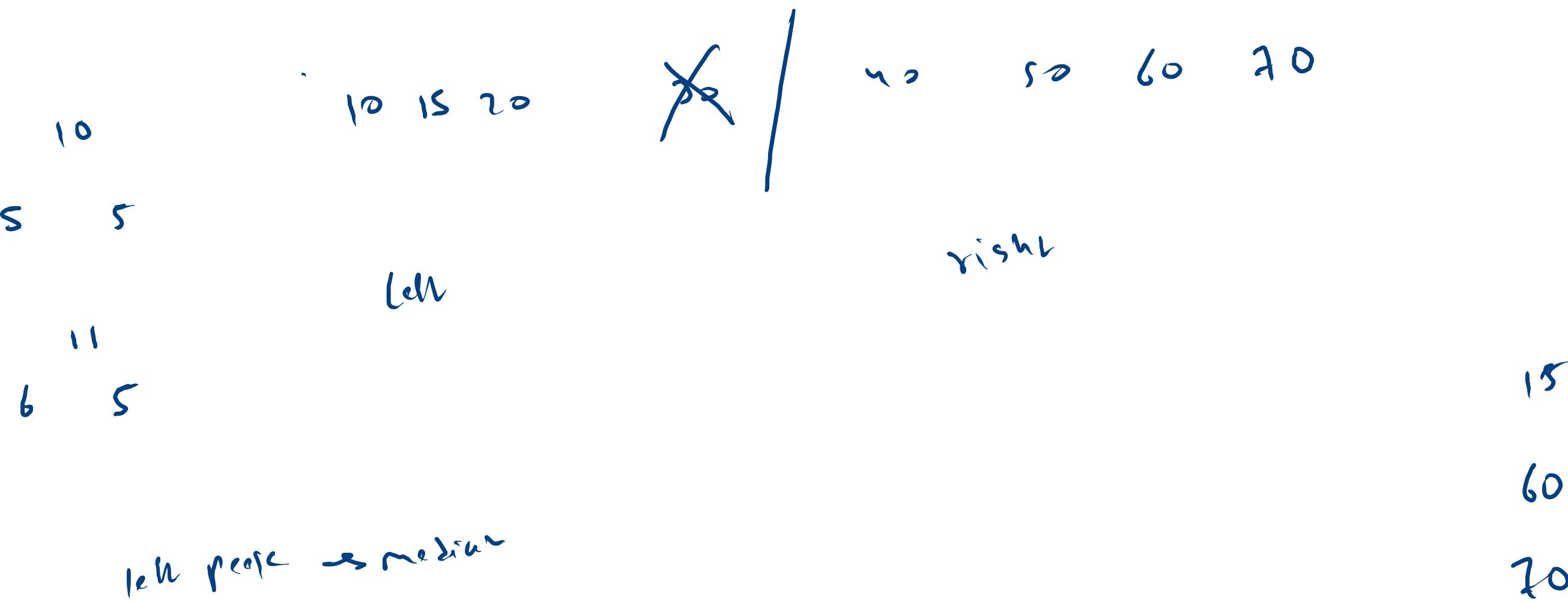
20

40

35

33
16





10 15 20 / 30 40

15

22
12

33
22

