Complexity: TRUE or FALSE

a) $n^2 \in O(n^3)$ True

$f(n) \in O(g(n))$ if $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)}$ is 0 or const (not ∞)

$$\lim_{n\to\infty} \frac{n^2}{n^3} = \lim_{n\to\infty} \frac{1}{n} = 0$$

b) $n^3 \in O(n^2)$ False

$$\lim_{n\to\infty} \frac{n^3}{n^2} = \lim_{n\to\infty} n = \infty$$

c) $2^{n+1} \in \Theta(2^n)$ True

$f(n) \in \Theta(g(n))$ if $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)}$ is a nonzero const (not ∞)

$$\lim_{n\to\infty} \frac{2^{n+1}}{2^n} = \lim_{n\to\infty} \frac{2^n \cdot 2}{2^n} = 2$$

d) $2^{2n} \in \Theta(2^n)$ False

$$\lim_{n\to\infty} \frac{2^{2n}}{2^n} = \infty$$

e) $n^2 \in \Theta(n^3)$ False

$$\lim_{n\to\infty} \frac{n^2}{n^3} = \lim_{n\to\infty} \frac{1}{n} = 0$$

f) $2^n \in \Theta(n!)$ True

$$\lim_{n\to\infty} \frac{2^n}{n!} = \lim_{n\to\infty} \frac{2}{n+1} = 0$$

$$\frac{a_{n+1}}{a_n} = \frac{2^{n+1}}{(n+1)!} \cdot \frac{n!}{2^n} = \frac{2}{n+1}$$

g) $\log_{10} n \in \Theta(\log_2 n)$ True

$$\lim_{n\to\infty} \frac{\log_{10} n}{\log_2 n} = \frac{\frac{\ln n}{\ln 10}}{\frac{\ln n}{\ln 2}} = \lim_{n\to\infty} \ln 2 = \ln 2$$

h) $O(n) + \Theta(n^2) = \Theta(n^2)$ True

upper bound $[0, n]$

tight bound (both lower and upper bound) $[n^2, n^2]$

$$\underbrace{\underset{\le n}{O(n)} + \underset{\le n^2}{\Theta(n^2)}}_{n^2} \in O(n^2)$$

$$\underbrace{\underset{\ge 0}{O(n)} + \underset{\ge n^2}{\Theta(n^2)}}_{n^2} \in \Omega(n^2)$$

$$= \Theta(n^2) \quad \in \Theta(n^2)$$

$\Theta(n) + O(n^2) = O(n^2)$ True

$$\underbrace{\le n \qquad \le n^2}_{\substack{+ \\ n^2}} \xrightarrow{\in} O(n^2) \Rightarrow \text{upper bound}$$

$$\underbrace{\ge n \qquad \ge 0}_{\substack{+ \\ n}} \rightarrow \Omega(n) \Rightarrow \text{lower bound}$$

$\Theta(n) + O(n^2) = \Omega(n)$ True

i) $O(n) + O(n^2) = O(n^2)$ True

$$\underbrace{\le n \qquad \le n^2}_{\substack{+ \\ n^2}} \Rightarrow \in O(n^2)$$

j) $O(n) + \Theta(n) = O(n)$ True

$$\underbrace{\le n \qquad \le n}_{\substack{+ \\ n}} \rightarrow \in O(n)$$

k) $(n+m)^2 \in O(n^2+m^2)$ True

$$n^2+2nm+m^2 \le n^2+m^2+2(n^2+m^2) = 3(n^2+m^2)$$

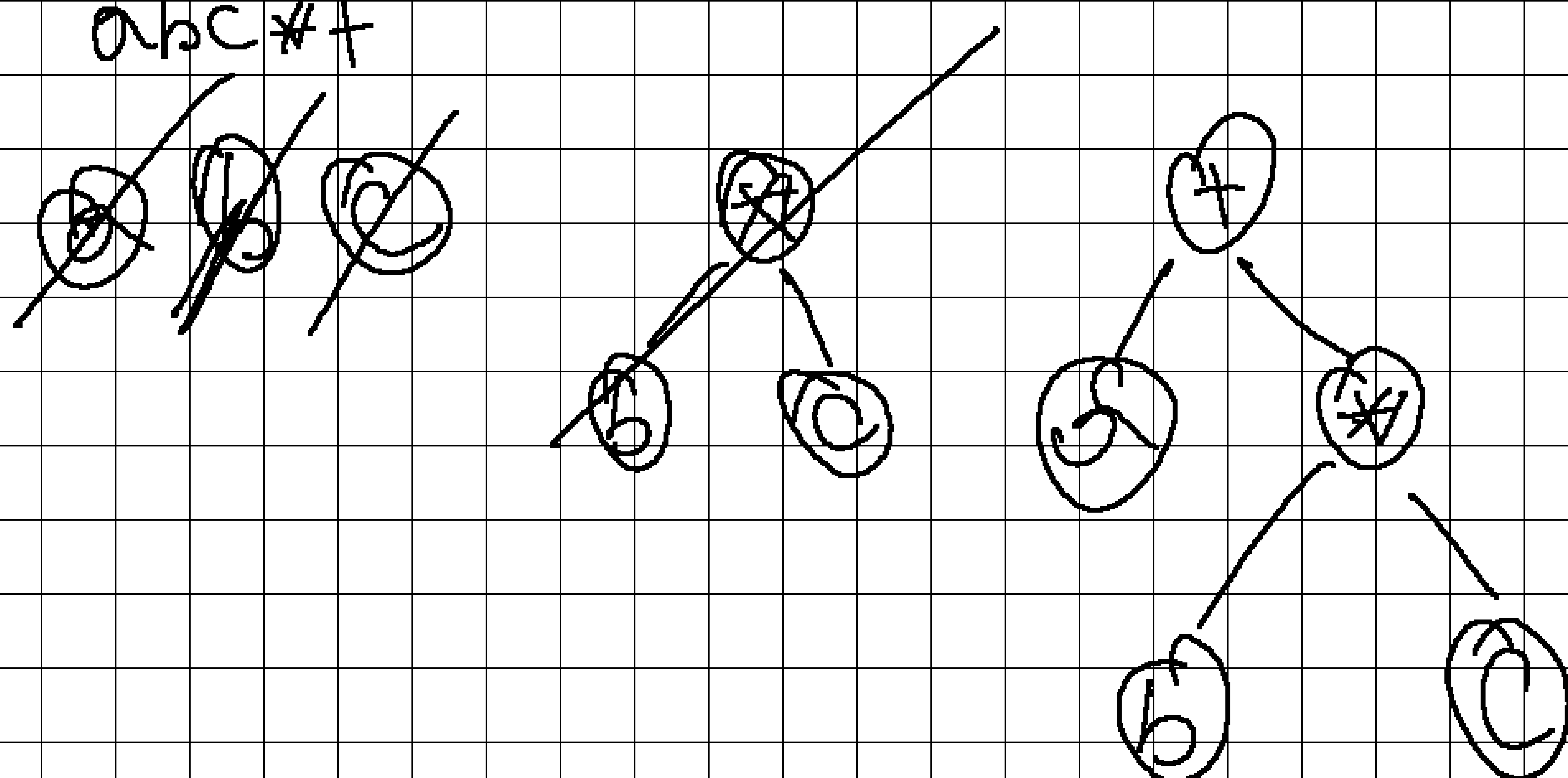
i) $3^n \in O(2^n)$ False

$$\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty} \left(\frac{3}{2}\right)^n = \infty$$

m) $\log_2 3^n \in O(\log_2 2^n)$ True

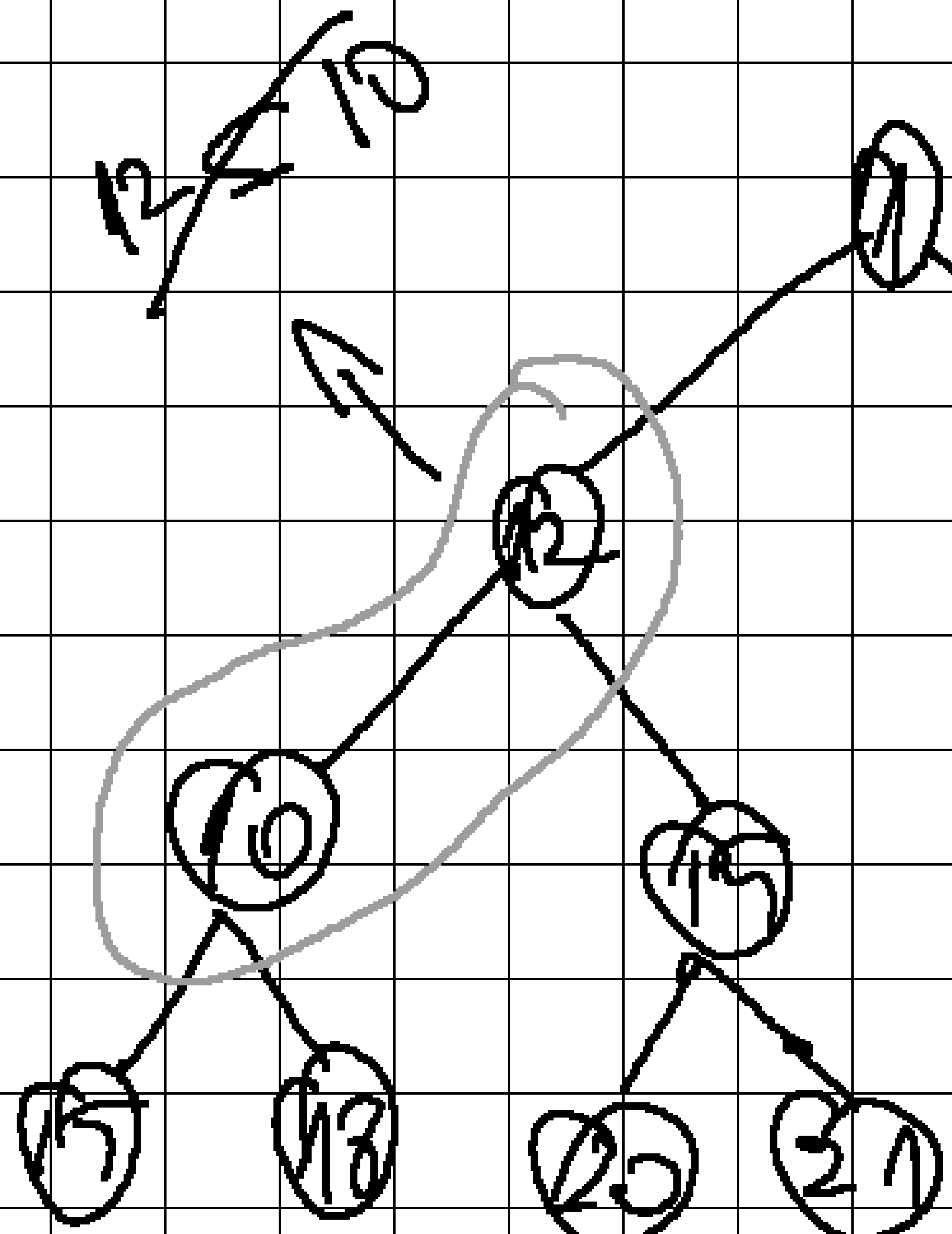
$$\lim_{n\to\infty} \frac{\log_2 3^n}{\log_2 2^n} = \lim_{n\to\infty} \frac{\cancel{n}\log_2 3}{\cancel{n}\log_2 2} = \frac{\log_2 3}{\log_2 2}$$

abc*+



2. a) Which of the following three arrays represents a binary heap? Justify why the other two arrays are not b. heaps

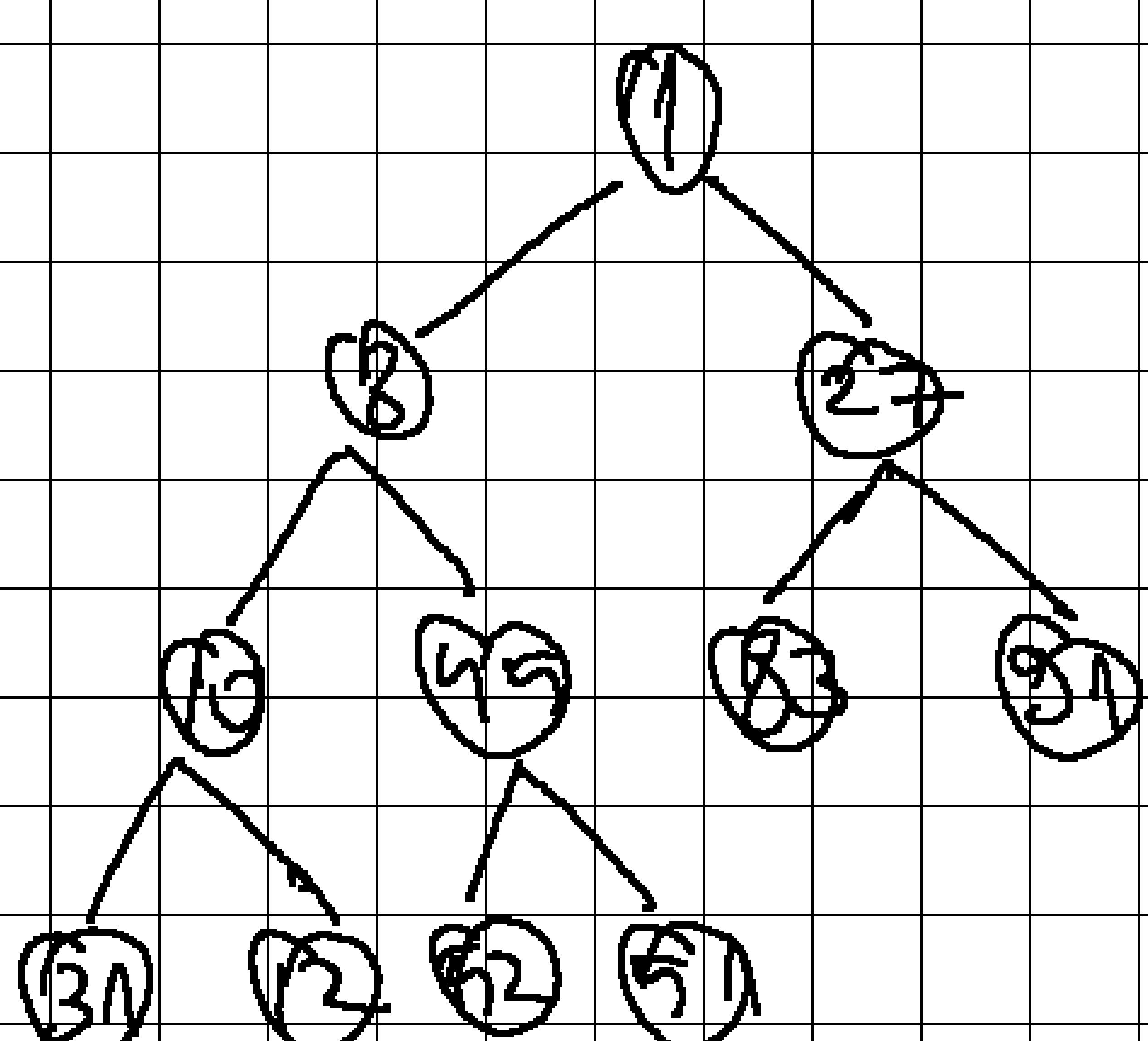
i) [1, 12, 23, 10, 15, 38, 45, 15, 18, 20, 21]



heap structure:
- every node has exactly 2 children, except for the last two levels where children are completed from left to right

heap property does not hold

ii) [1, 8, 27, 10, 45, 83, 91, 31, 12, 52, 51]



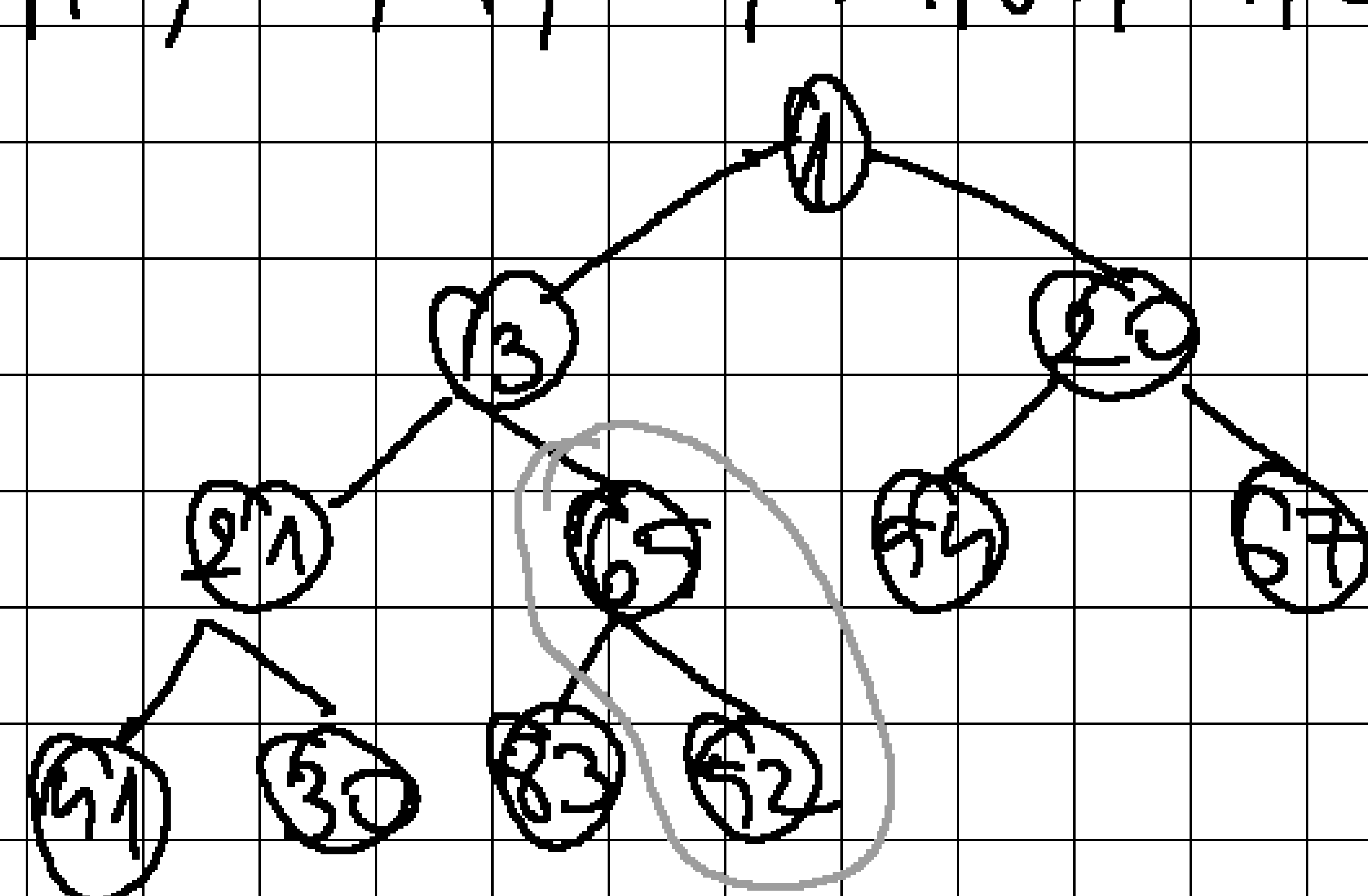
heap structure holds

heap property:
- we have a min-heap

$a_i \le a_{2*i}$

$a_i \le a_{2*i+1}$

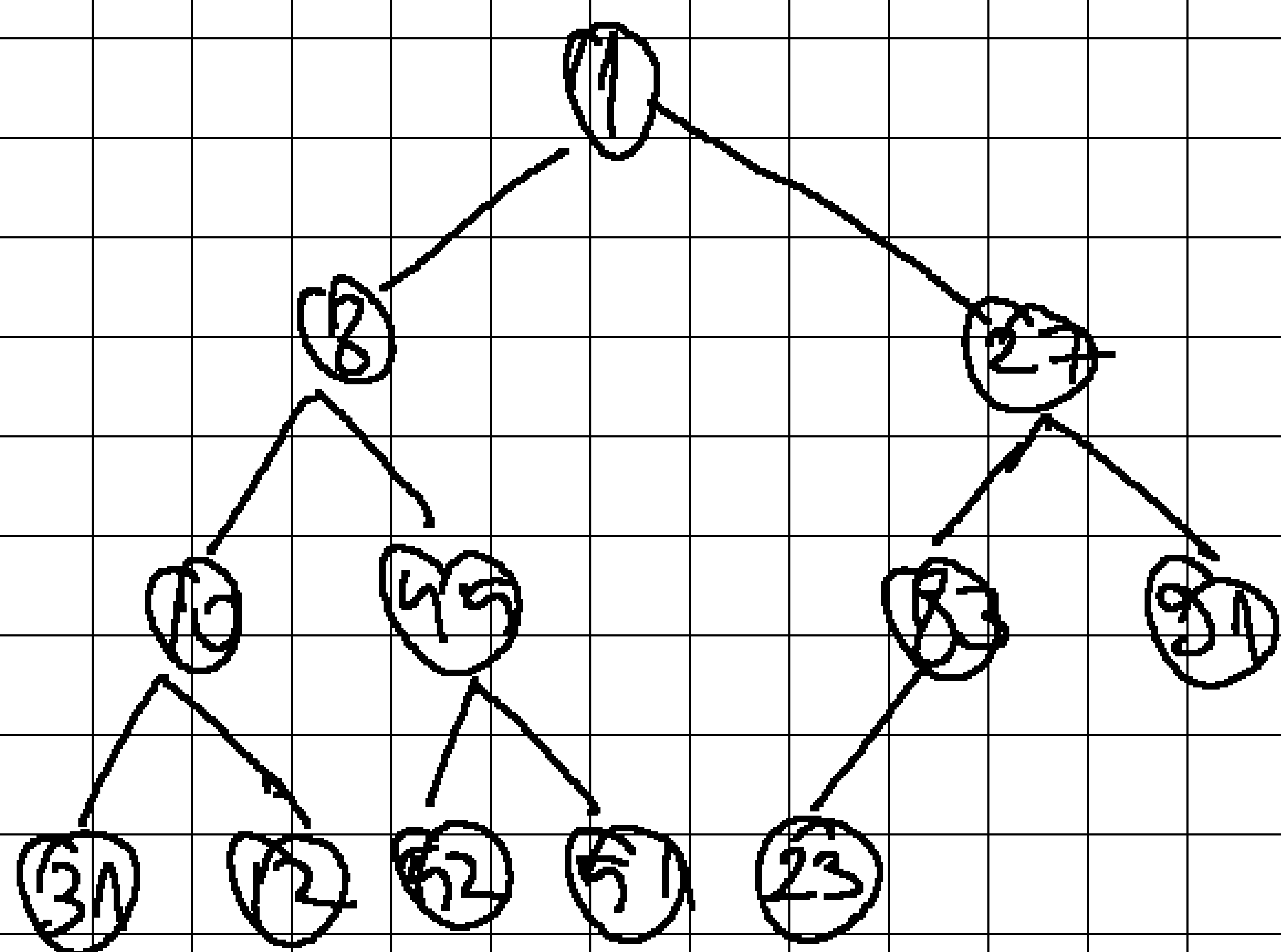
iii) [1, 13, 20, 21, 65, 94, 67, 41, 30, 83, 52]



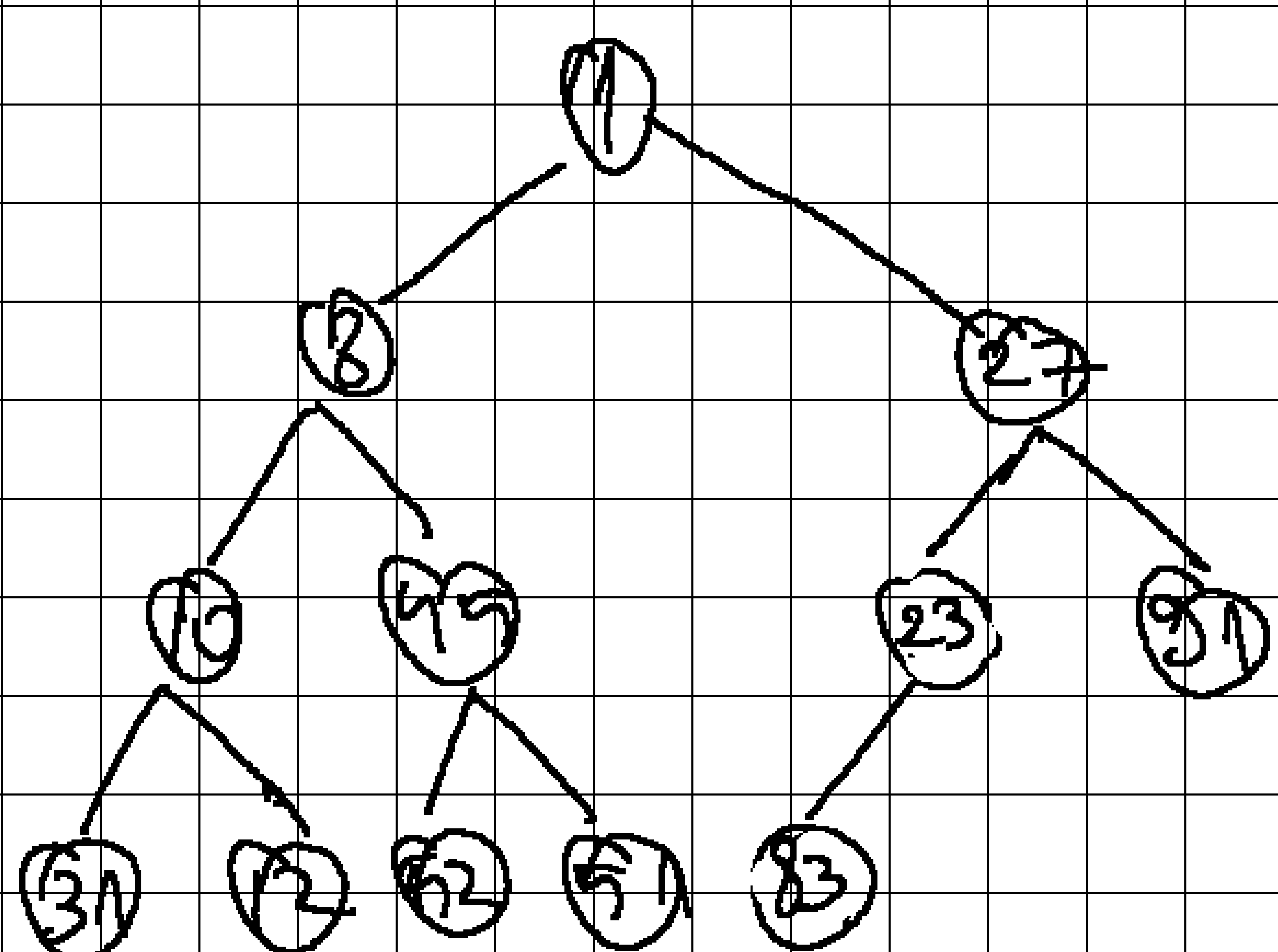
heap structure holds

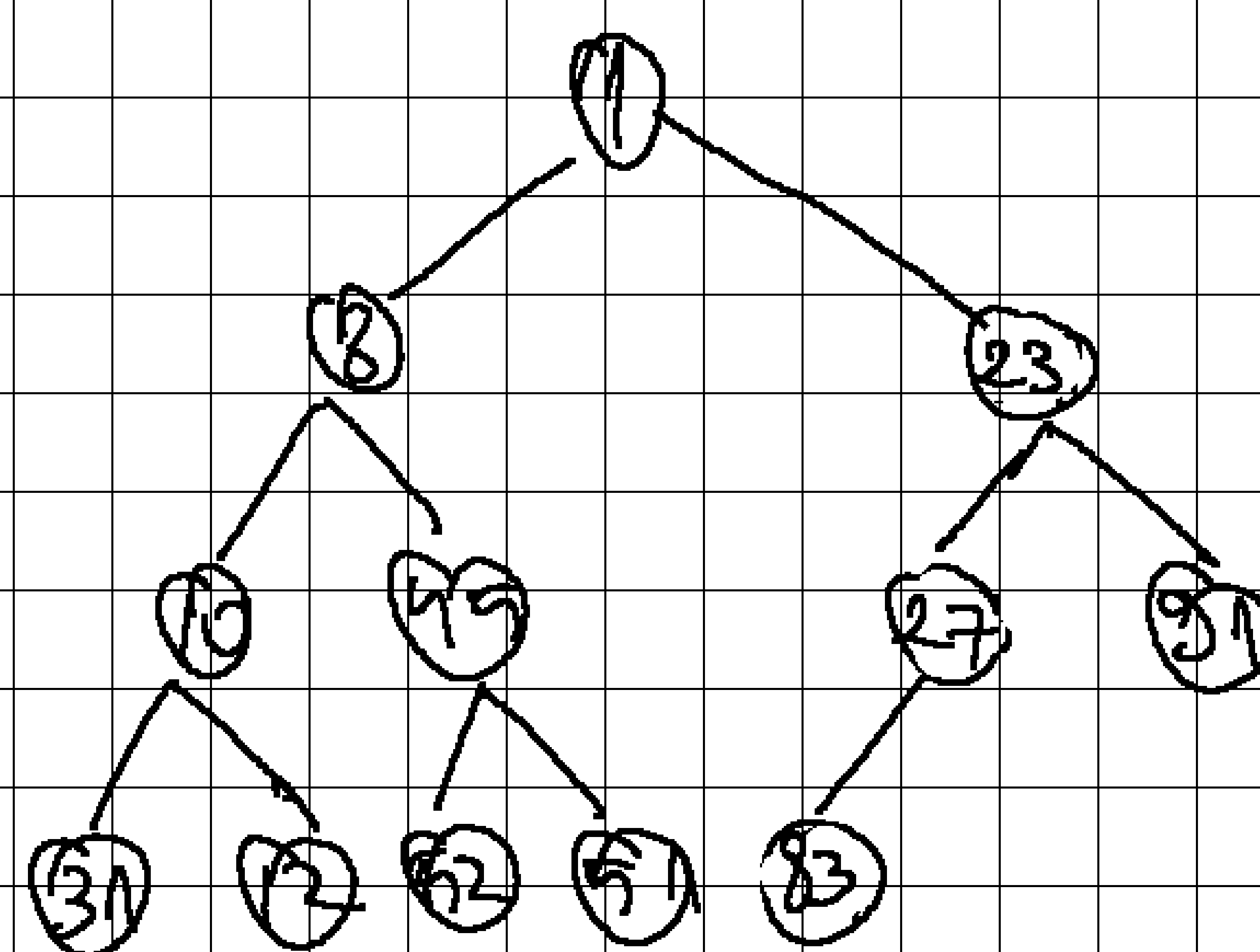
heap property does not hold

To the selected heap add the number 23. Show the tree form after the insertion. Perform two delete operations on the heap and show the result.

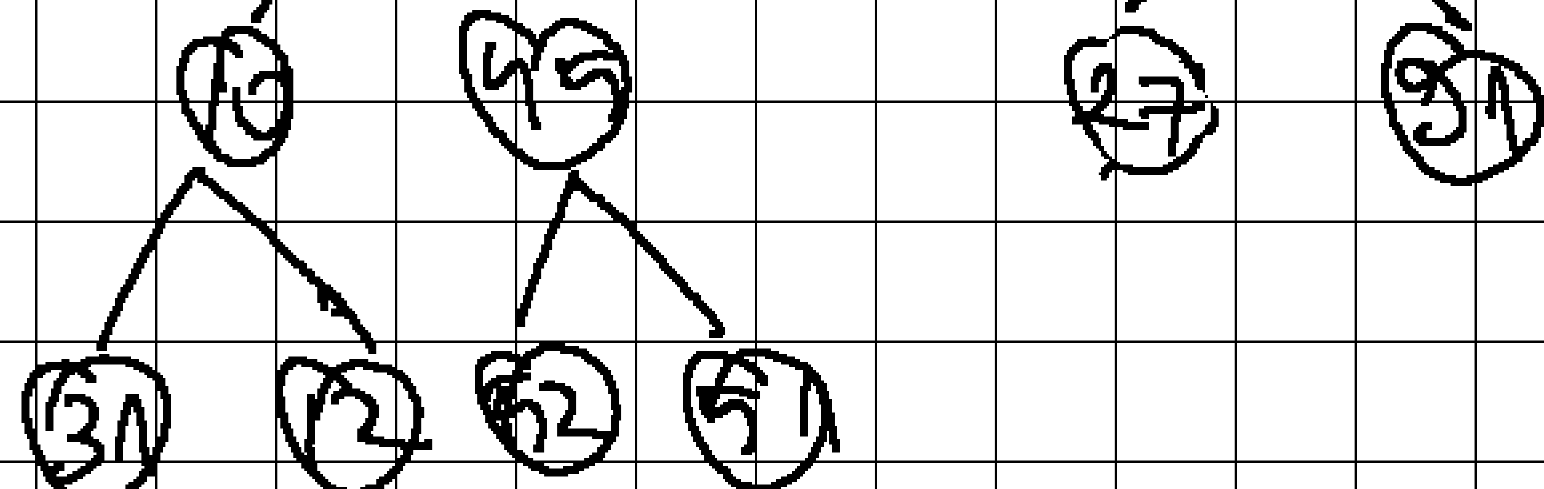


we have to do a bubble up

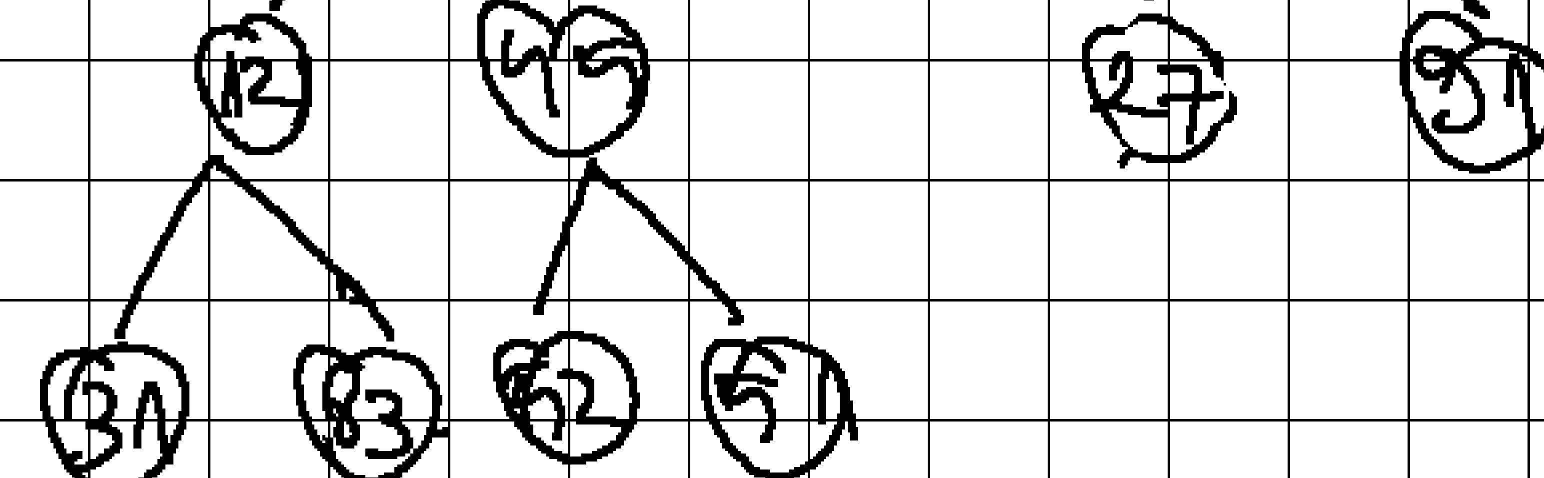




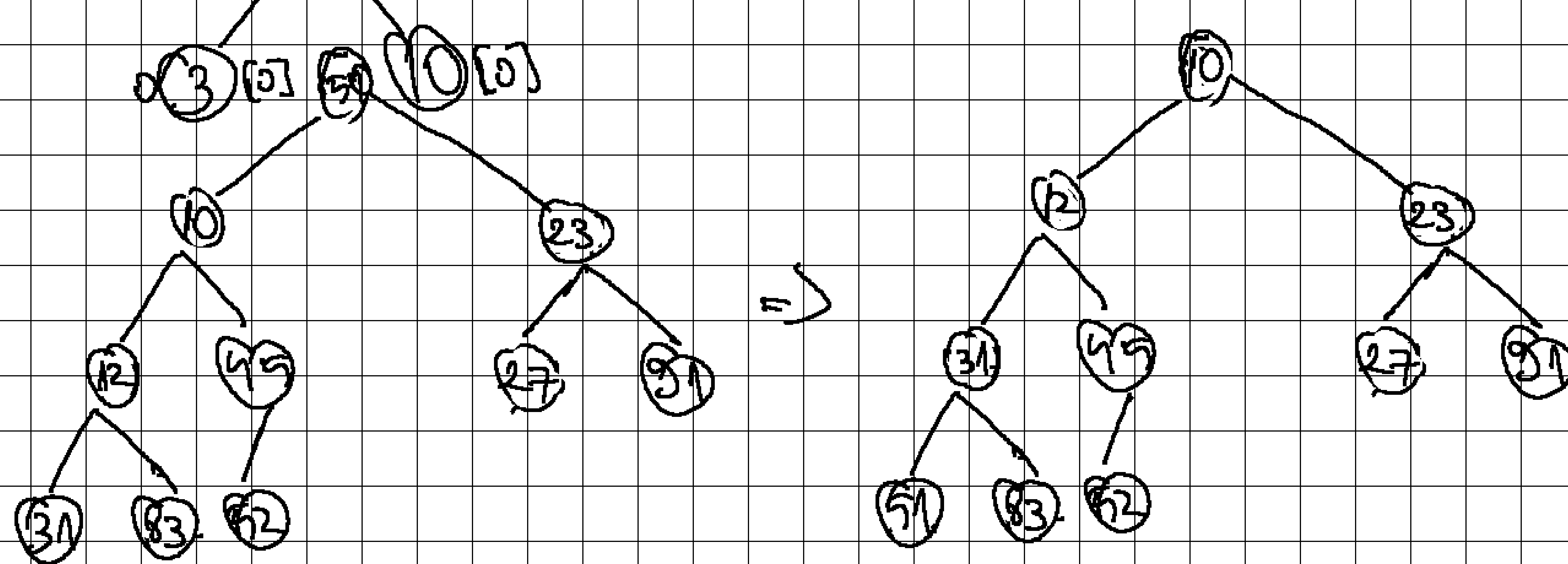
Delete



=>



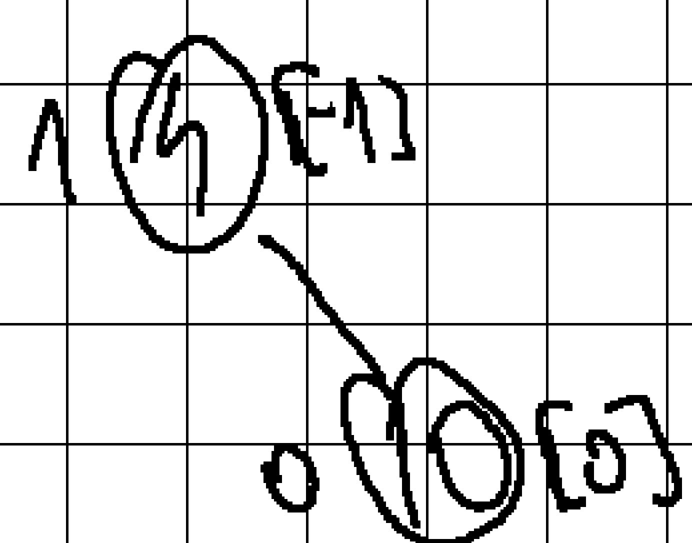
we have to do a bubble down



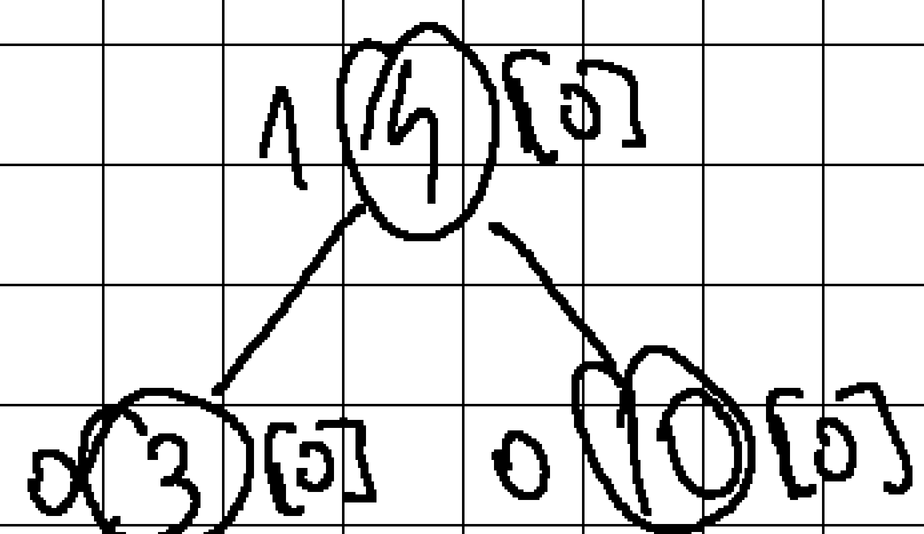
we have to do a bubble down

b) I.

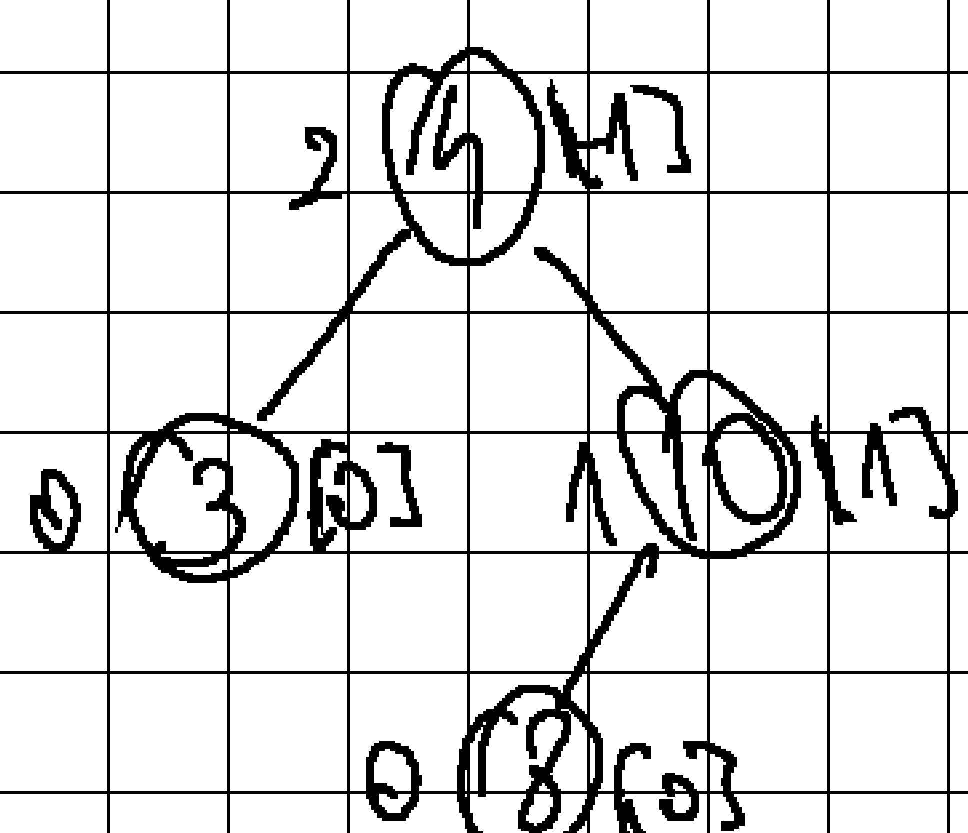
II.



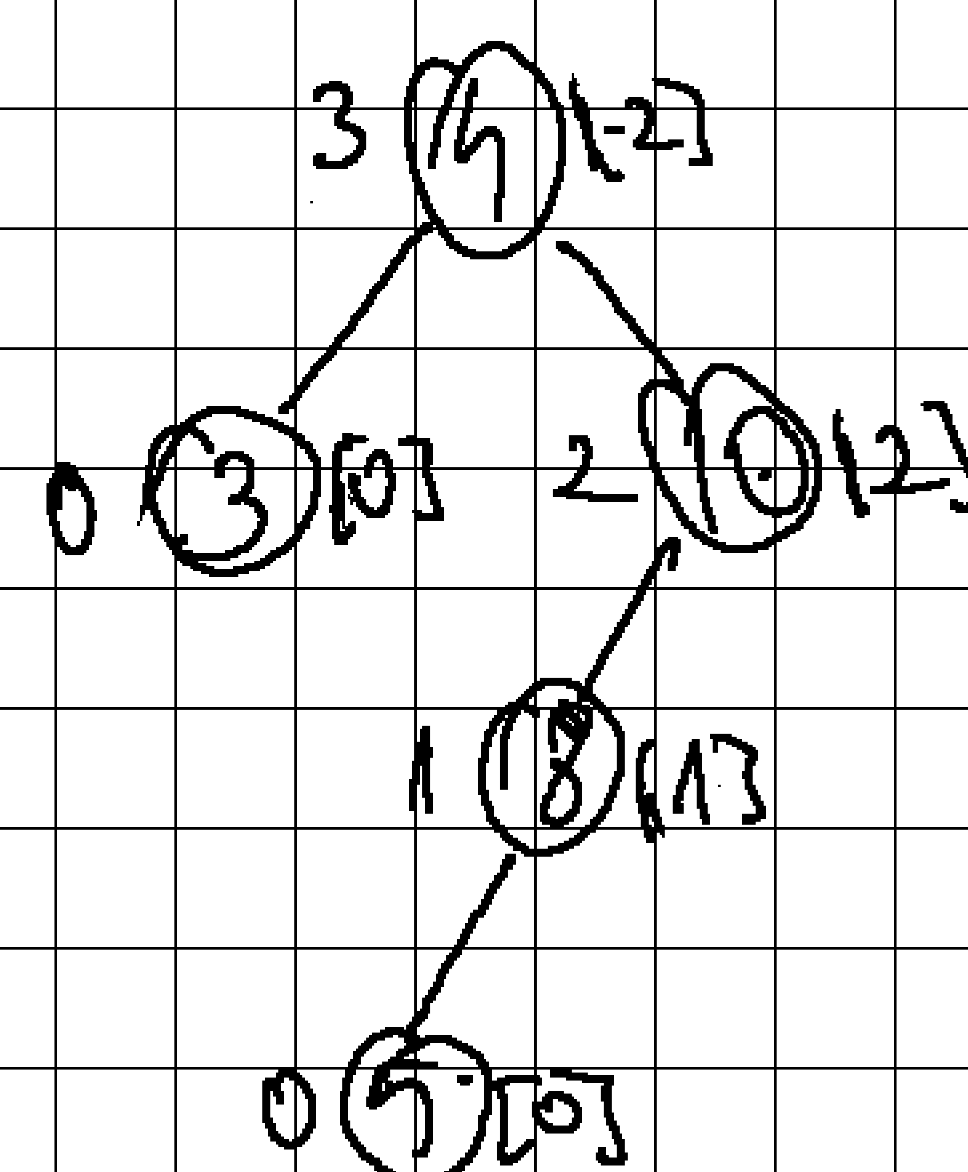
III.



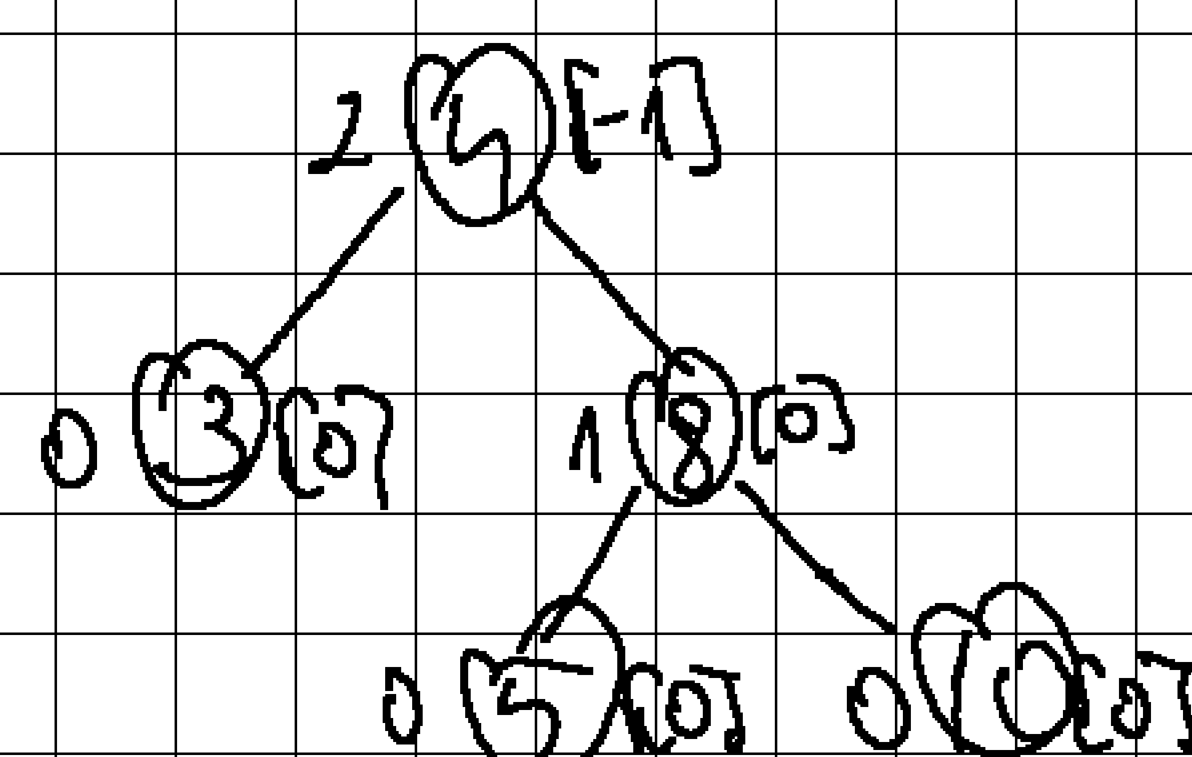
IV



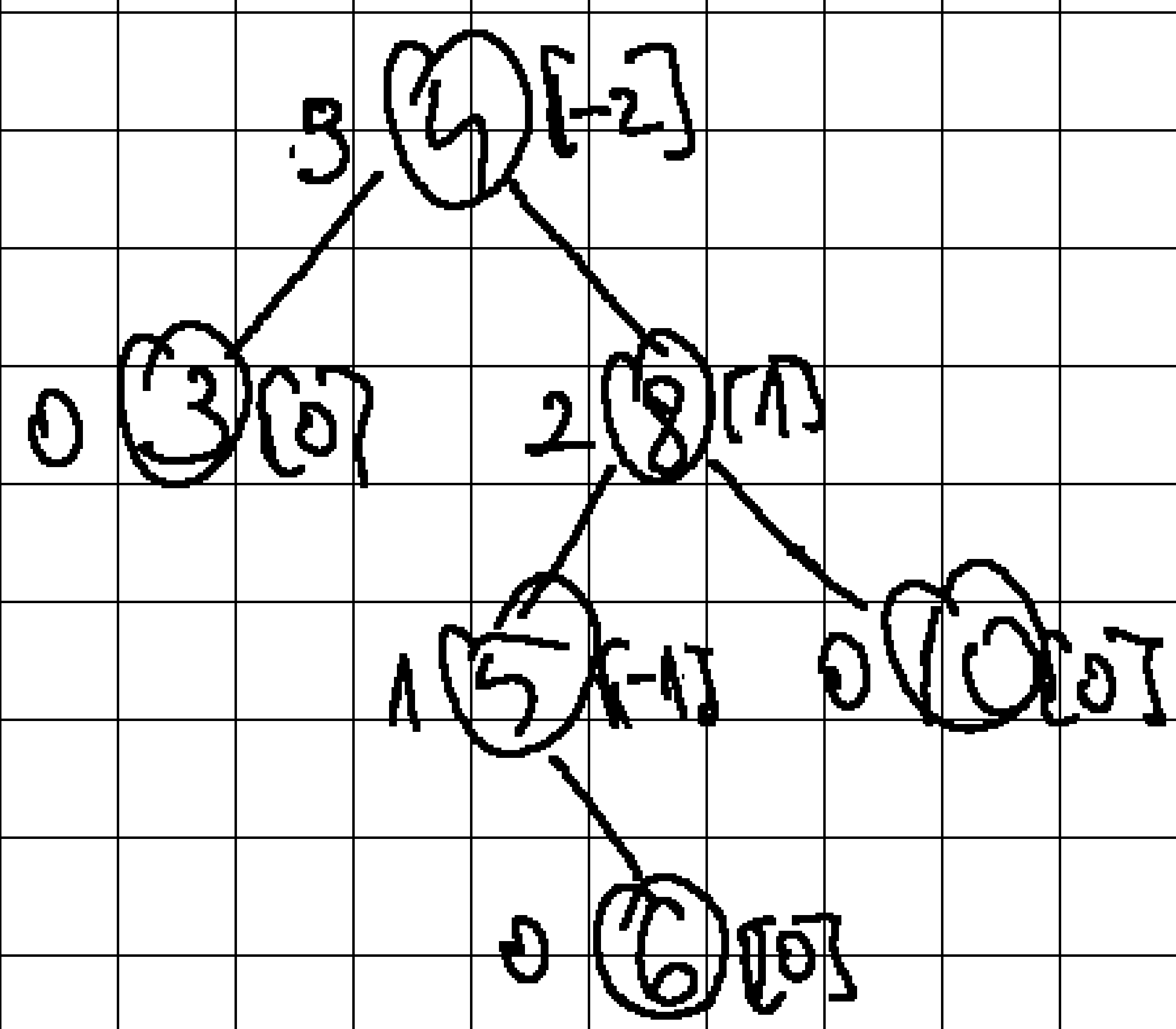
IV



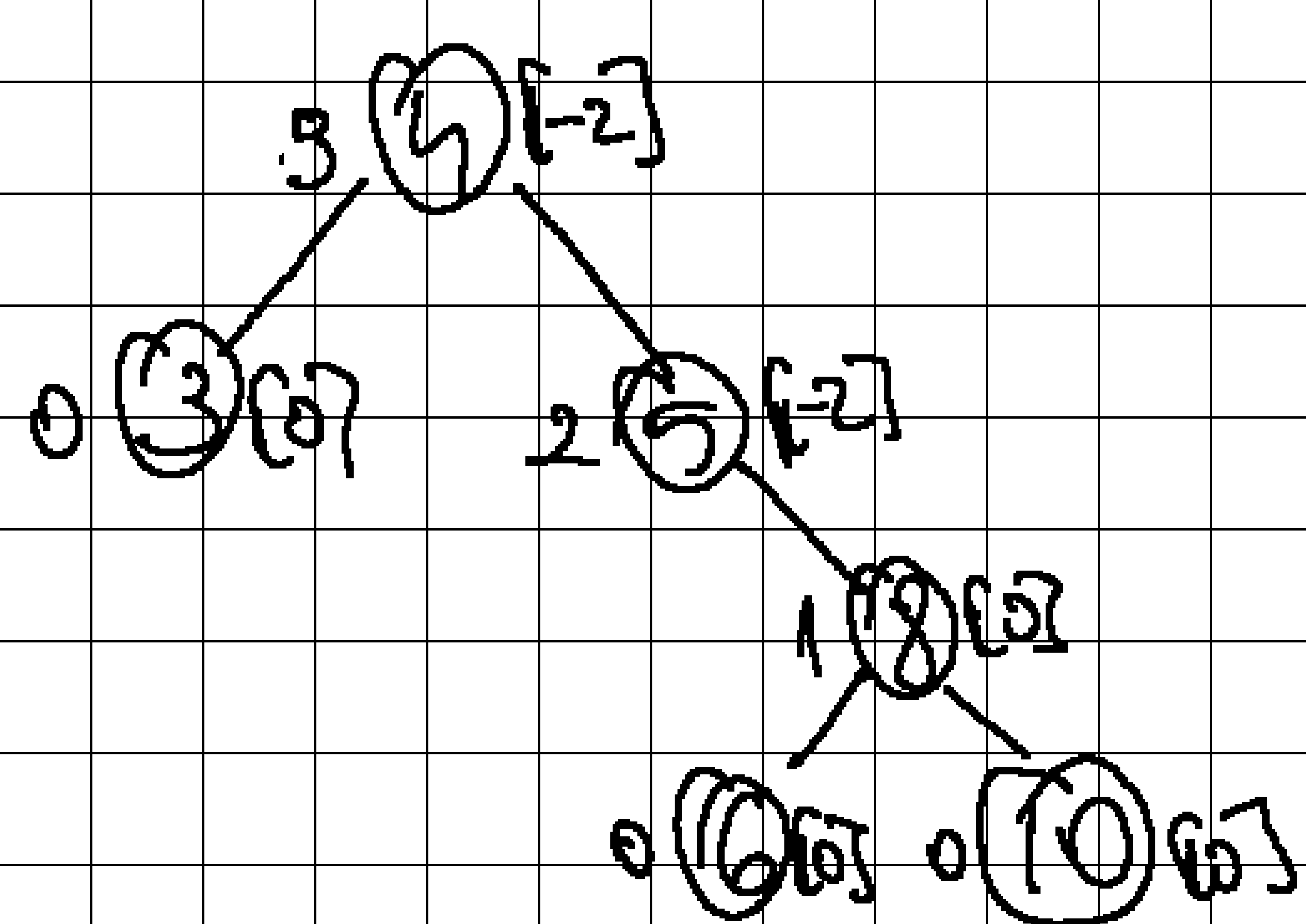
imbalance at node 10
(heavy left => rotation to right)

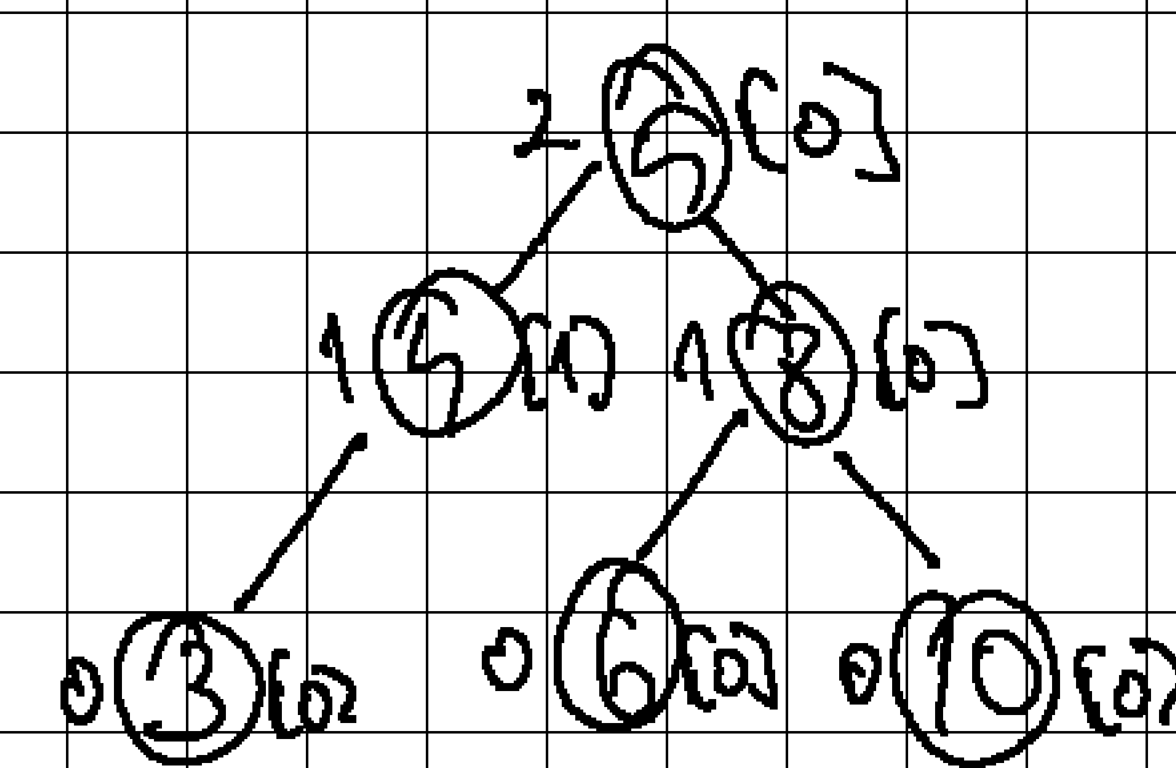


V

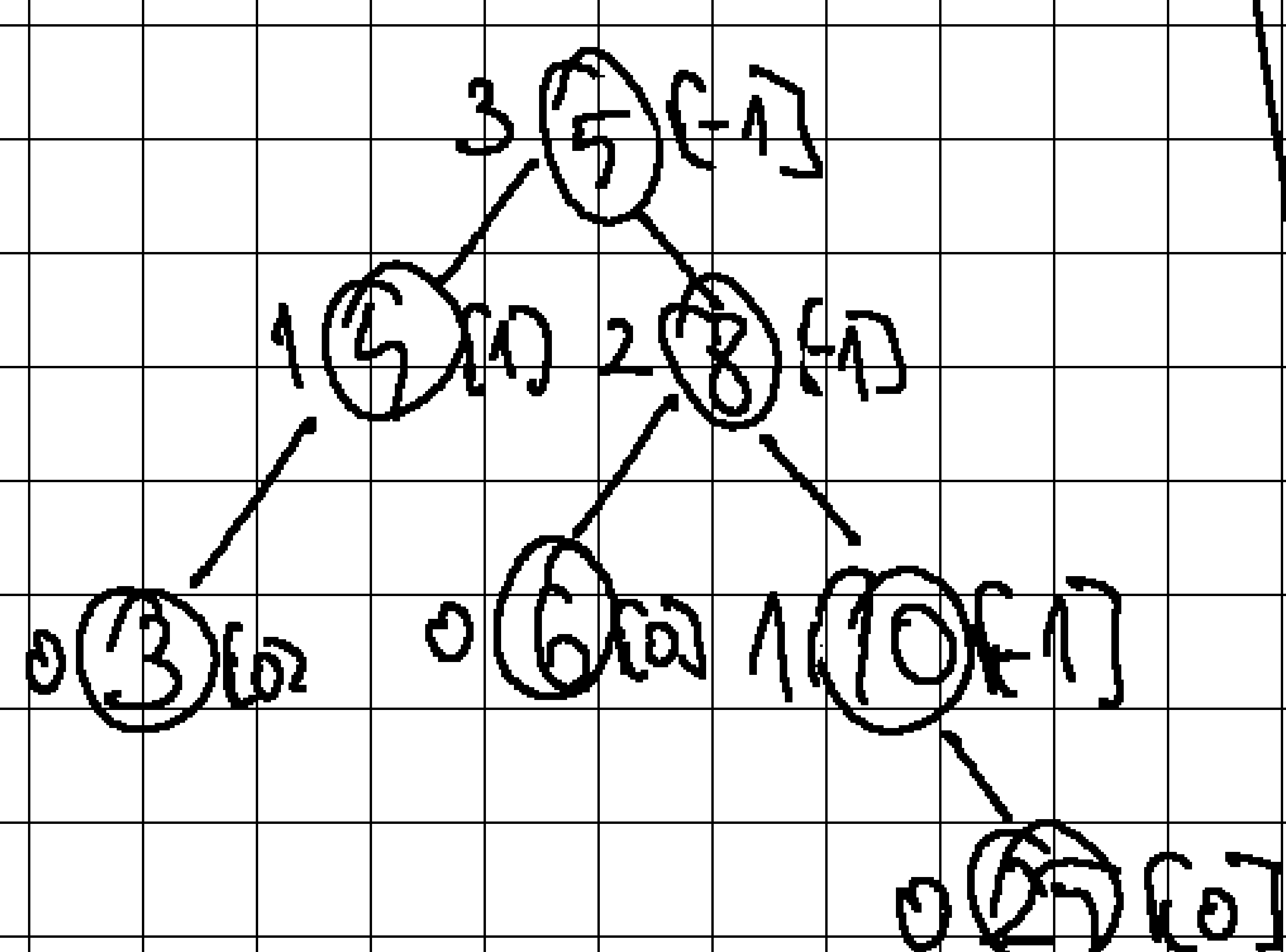


imbalance at node 4 (need a double left rotation)

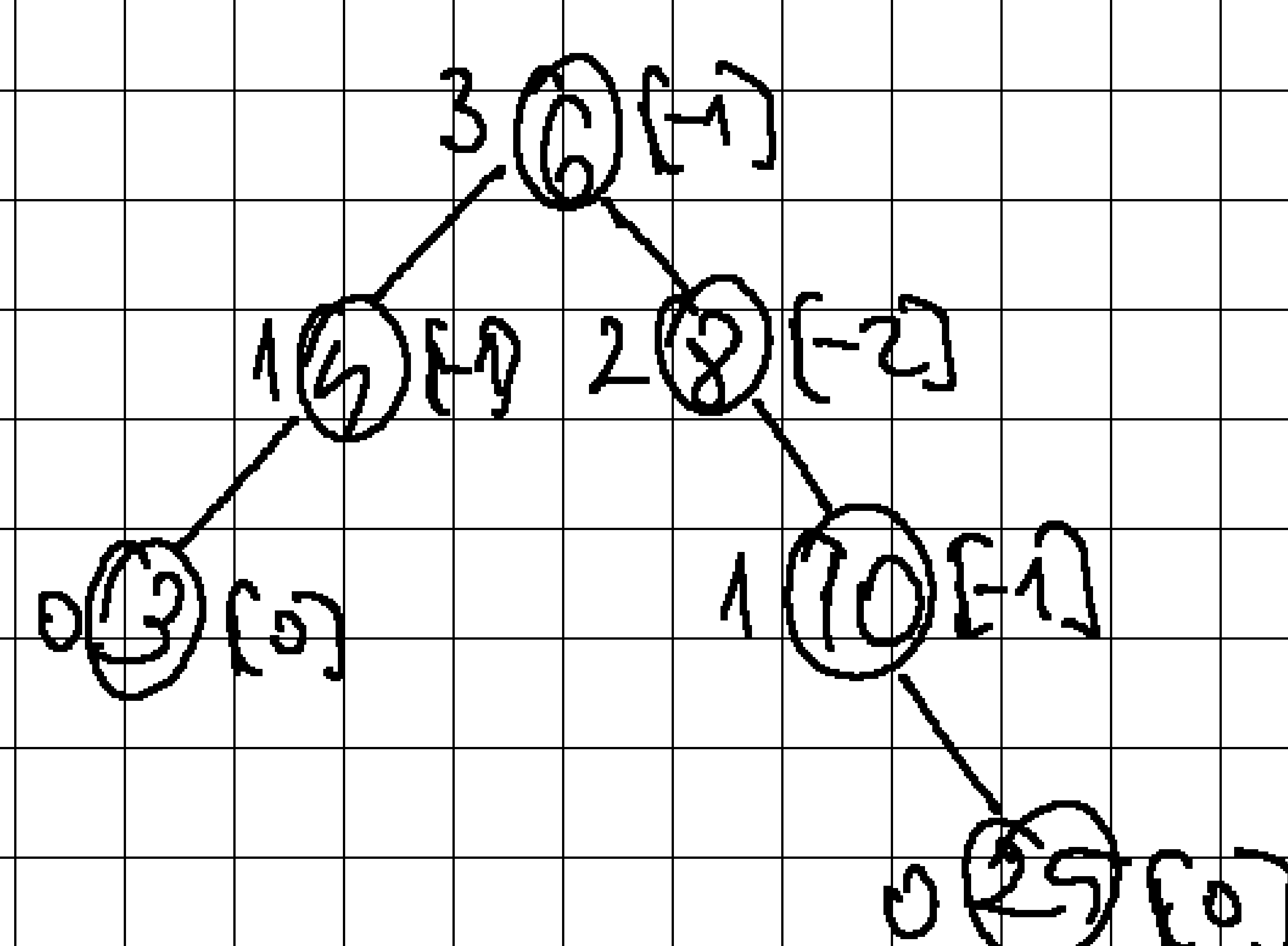




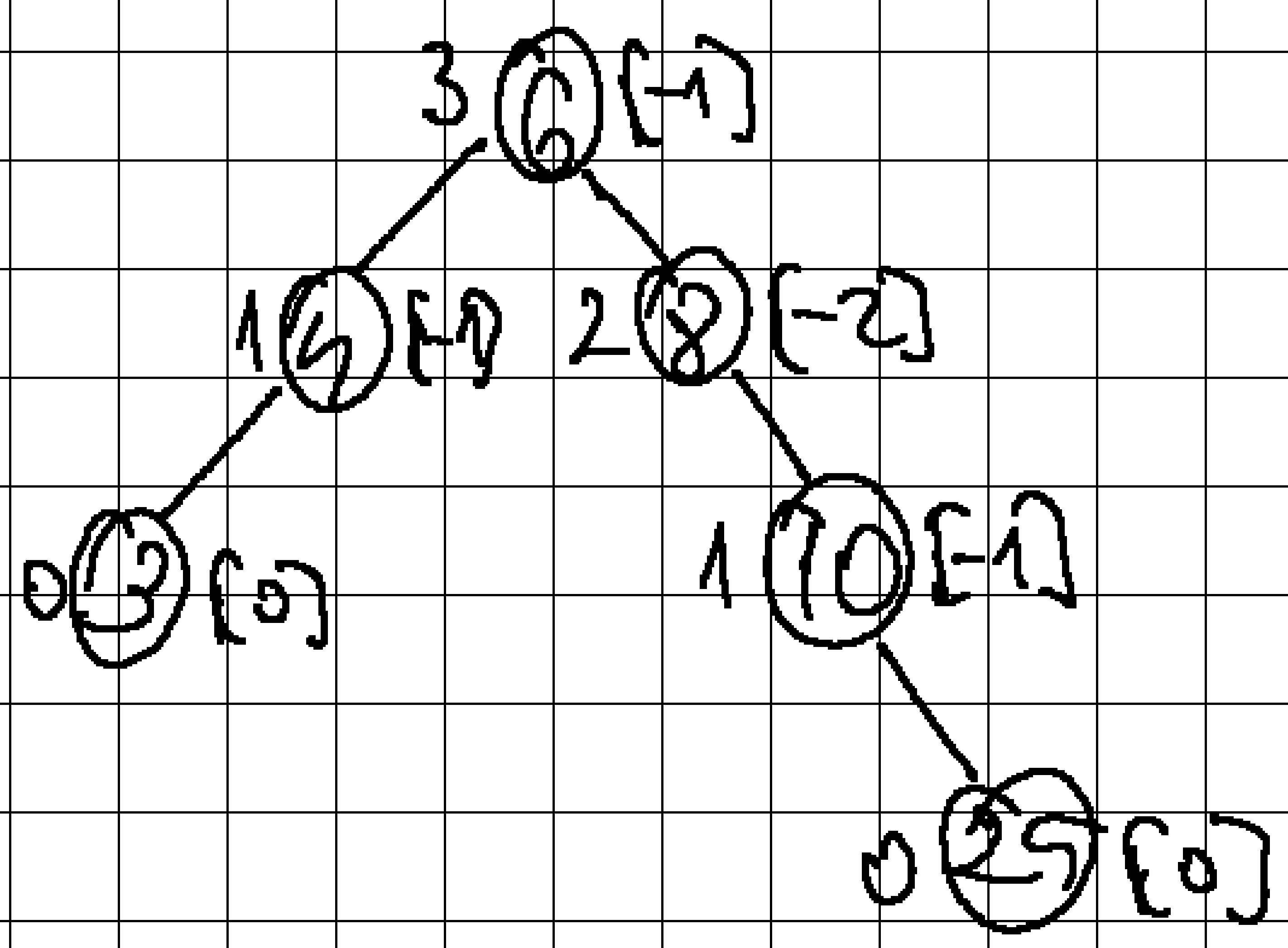
VI



Remove node 5

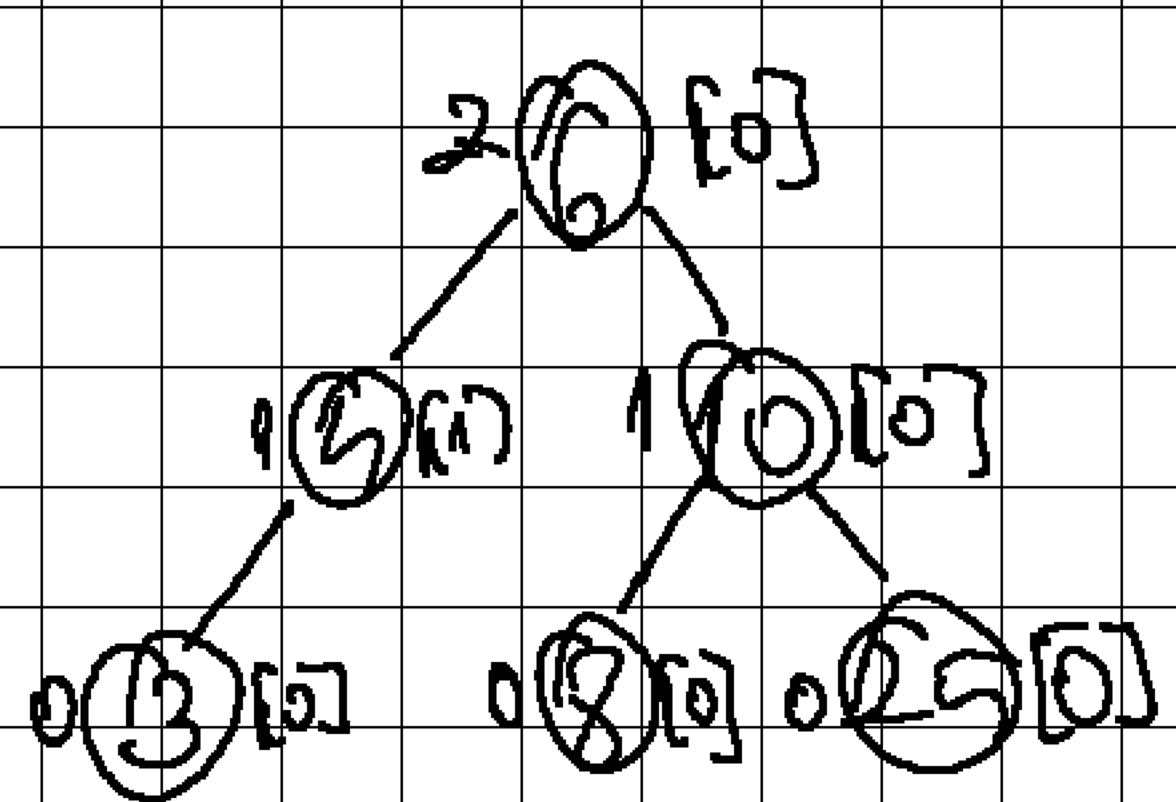


Remove node 5



Imbalance at node 8

We need a left rotation



c) $m = 7$

i)

values	13	17	6	24	3
hash	6	3	6	3	3

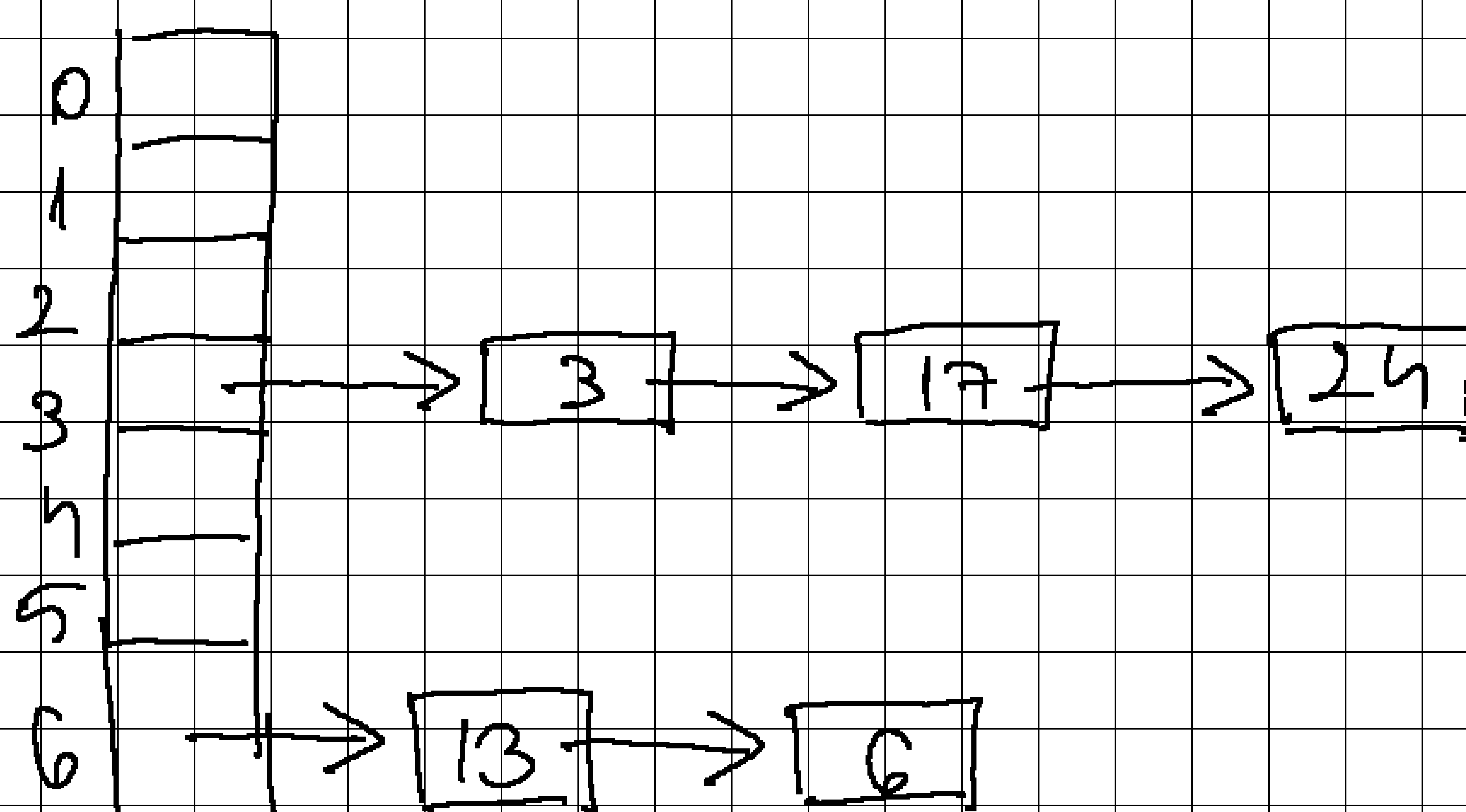
$h(13) = 13 \% 7 = 6$

$h(17) = 17 \% 7 = 3$

$h(6) = 6 \% 7 = 6$

$h(24) = 24 \% 7 = 3$

$h(3) = 3 \% 7 = 3$



ii)

values	13	17	6	24	3
hash	4	5	6	3	4

$h_2(13) = 7 - (13 \% 5) = 4$

$h_2(17) = 7 - (17 \% 5) = 5$

$h_2(6) = 7 - (6 \% 5) = 6$

$h(24) = 7 - (24 \% 5) = 3$

$h(3) = 7 - (3 \% 5) = 4$

hashtable

0	1	2	3	4	5	6
	24		17	3	6	13

3. a) $h(9) = 9 \% 9 = 0$

$h(14) = 14 \% 9 = 5$

$h(4) = 4 \% 9 = 4$ X $\Rightarrow$ 4 should be on pos 6

$h(18) = 18 \% 9 = 0 + 1 = 1$

b) $h(12) = 12 \% 9 = 3$

$h(3) = 3 \% 9 = 3 + 1 = 4$

$h(14) = 14 \% 9 = 5$

$h(18) = 18 \% 9 = 0$ X $\Rightarrow$ 18 should be on pos 1 not 0

$\boxed{c)}$ $h(12) = 12 \% 9 = 3$

$h(14) = 14 \% 9 = 5$

$h(3) = 3 \% 9 = 3 + 1 = 4$

$h(9) = 9 \% 9 = 0$

$h(4) = 4 \% 9 = 4 + 1 = 5 + 1 = 6$

$h(18) = 18 \% 9 = 0 + 1 = 1$

$h(21) = 21 \% 9 = 3 + 1 = 4 + 1 = 5 + 1 = 6 + 1 = 7$

$\boxed{d)}$ $h(9) = 9\%9 = 0$

$h(12) = 12\%9 = 3$

$h(14) = 14\%9 = 5$

$h(3) = 3\%9 = 3+1 = 4$

$h(4) = 4\%9 = 4+1 = 5+1 = 6$

$h(21) = 21\%9 = 3+1 = 4+1 = 5+1 = 6+1 = 7$

$h(18) = 18\%9 = 0+1 = 1$

e) $h(12) = 12\%9 = 3$

$h(9) = 9\%9 = 0$

$h(18) = 18\%9 = 0+1 = 1$

$h(3) = 3\%9 = 3+1 = 4$

$h(14) = 14\%9 = 5$

$h(21) = 21\%9 = 3+1 = 4+1 = 5+1 = 6$ X $\Rightarrow$ it should be on pos 7

2. 45 - 3 * 5 4 - * 3 +

Stack [$\cancel{4}$, $\cancel{5}$, $\cancel{1}$, $\cancel{3}$, $\cancel{3}$, $\cancel{5}$, $\cancel{4}$, $-\cancel{1}$, $-\cancel{3}$, $\cancel{3}$]

$5-4=1$

$1*3=3$

$4-5=-1$

$3*(-1)=-3$

$-3+3=0$

c) something between 5 and -5

3)

$\boxed{a)}$ $\log_{10} 2^N \in O(N)$ True

$$\lim_{N\to\infty} \frac{\log_{10} 2^N}{N} = \lim_{N\to\infty} \frac{\cancel{N}\log_{10} 2}{\cancel{N}} = \log_{10} 2$$

$\boxed{b)}$ $\log_{10} 2^N \in \Omega(N)$ True

$$\lim_{N\to\infty} = \log_{10} 2$$

c) $\log_{10} 2^N \in \Omega(N^2)$ False

$$\lim_{N\to\infty} \frac{\log_{10} 2^N}{N^2} = \lim_{n\to\infty} \frac{\log_{10} 2}{N} = 0$$

d) $\log_{10} 2^N \in O(\log_2 N)$

grows much faster to infinity then $\log_2 N$

$$\lim_{n\to\infty} \frac{\log_{10} 2^N}{\log_2 N} = \log_{10} 2 \lim_{N\to\infty} \frac{N}{\log_2 N} = \infty$$

e) $\log_{10} 2^N \in \Theta(N^2)$ False

$$\lim_{N\to\infty} \frac{\cancel{N}\log_{10} 2}{N^{\cancel{2}}} = 0$$

Exam: Sn:

2. 1. a) Pop from a stack on an array with N elems

The worst case complexity is $O(1)$ because popping involves removing the top element and that does not depend on the number of elems.

b) Find in a hash with open addressing (linear probing) with N elements, tablesize $= N^2$

The worst case complexity is $O(N)$, when all the elements hash to the same index, requiring a linear probing through N elements

c) insert in a hash with separate chaining, N elements tablesize: N^2

Worst case complexity is $O(1)$ because if the element hashes to an empty pos we just add it to the table, if it's occupied we add it to the begining of the list.

d) $F(n) = \log_2(\log_2(N+N)) + (\log_2 N)^2$

$\log_2(N+N) = \log_2 2N = \log_2 2 + \log_2 N = 1 + \log_2 N$

$F(n) = \log_2(\log_2 N) + \underbrace{(\log_2 N)^2}_{\text{grows much faster so is dominant}} \Rightarrow O((\log_2 N)^2)$

2. $m = 7$

$c_1 = 3 \quad c_2 = 2$ $\qquad h'(k) = 44 \% 7 = 2$

$h(k) = (h'(k) + c_1 * i + c_2 * i^2) \% m$

$i=0 \Rightarrow h(44) = (2+3*0+2*0) \% 7 = 2$

$i=1 \Rightarrow h(44) = (2+3*1+2*1) \% 7 = 7 \% 7 = 0$

$i=2 \Rightarrow h(44) = (2+3*2+2*4) \% 7 = 16 \% 7 = 2$

$i=3 \Rightarrow h(44) = (2+3*3+2*9) \% 7 = 29 \% 7 = 1$

$i=4 \Rightarrow h(44) = (2+3*4+2*16) \% 7 = 46 \% 7 = 4$

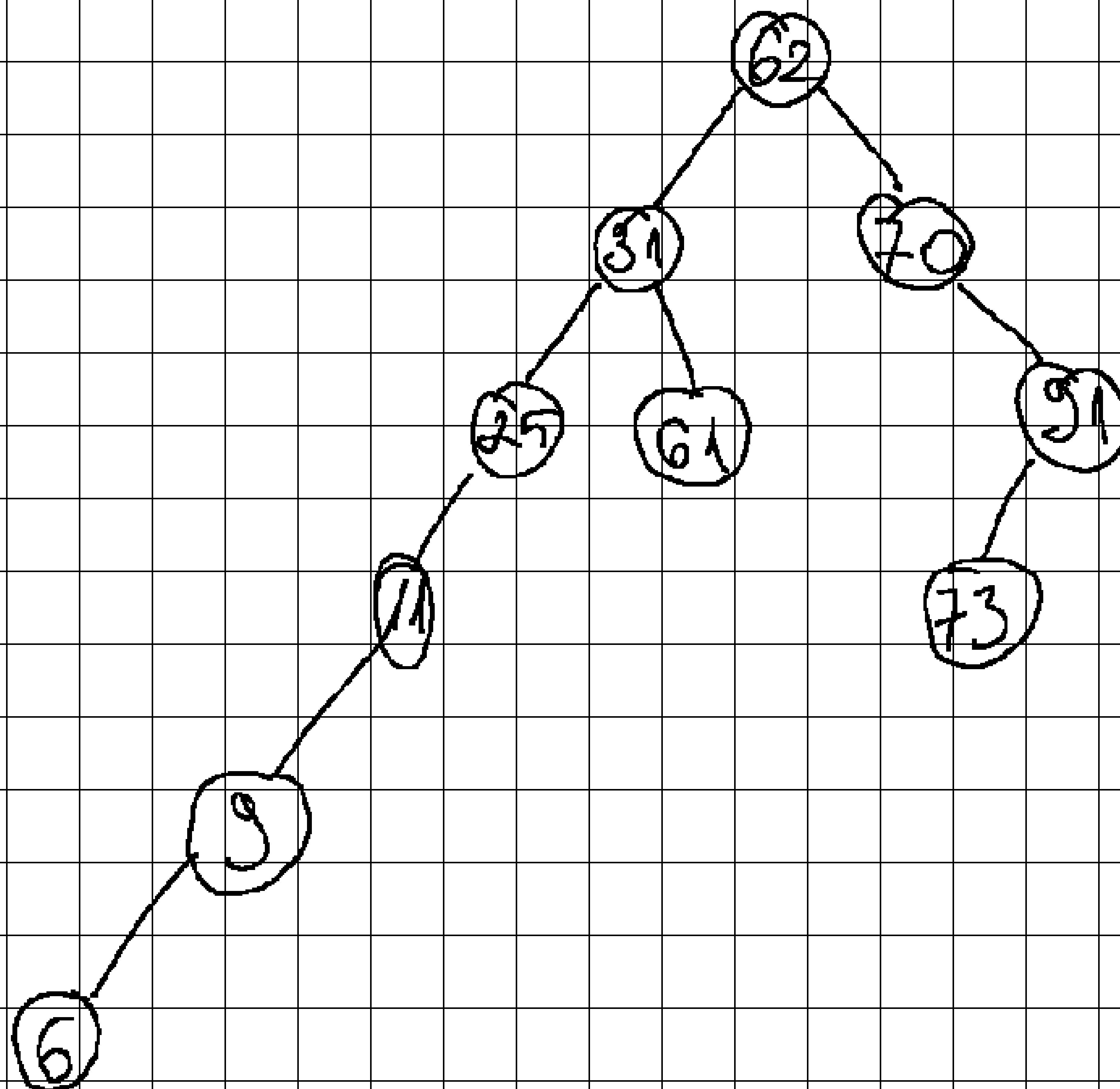
$i=5 \Rightarrow h(44) = (2+3*5+2*25) \% 7 = 67 \% 7 = 4$

$i=6 \Rightarrow h(44) = (2+3*6+2*36) \% 7 = 92 \% 7 = 1$

The probe sequence is $\langle 2, 0, 2, 1, 4, 4, 1 \rangle$

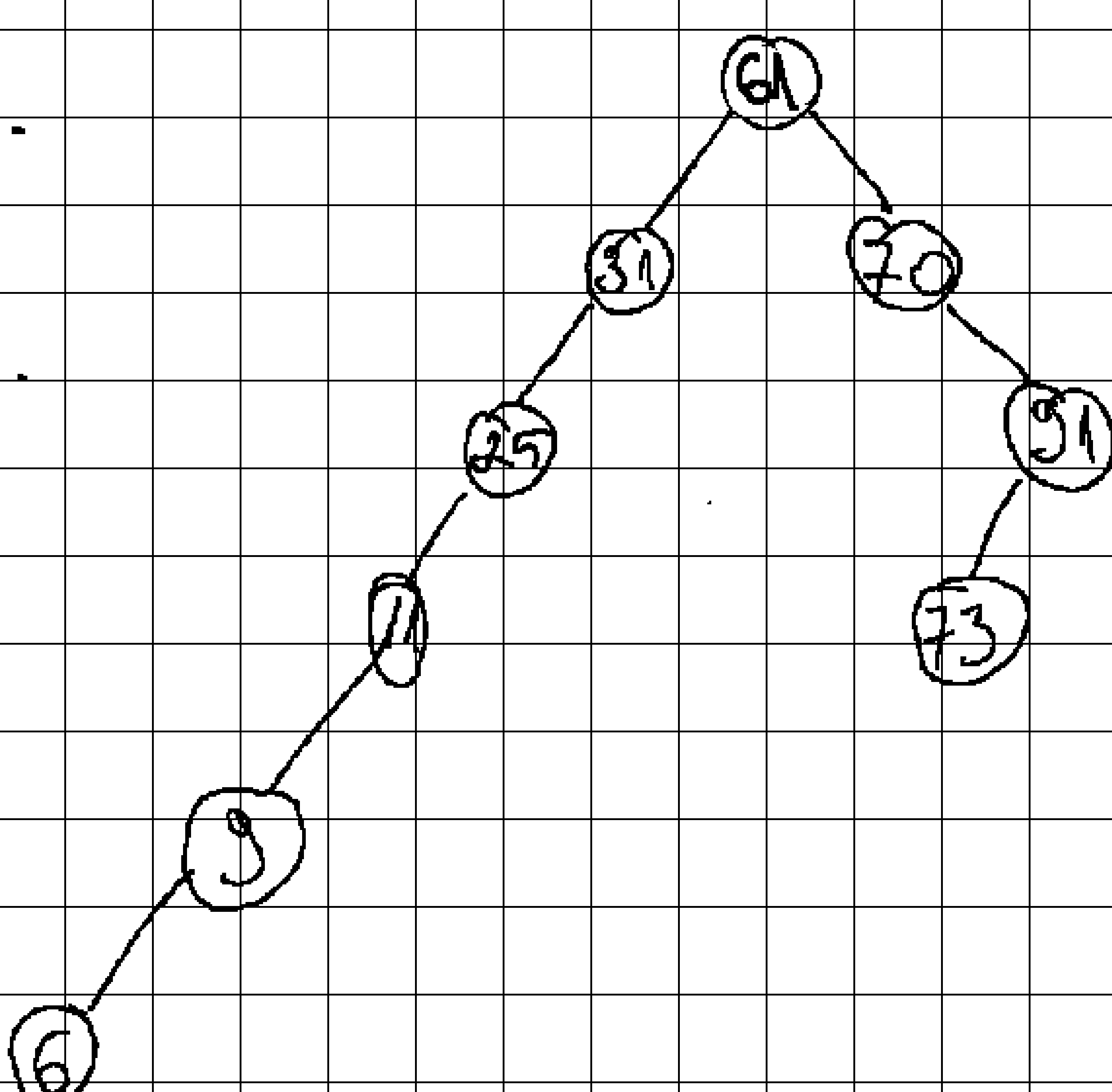
- the hash function is bad because
  - the sequence has repeated values (2, 4, 1) before covering all possible positions
  - a good hash function should avoid clustering and cover all table positions before repeating other slots

3.



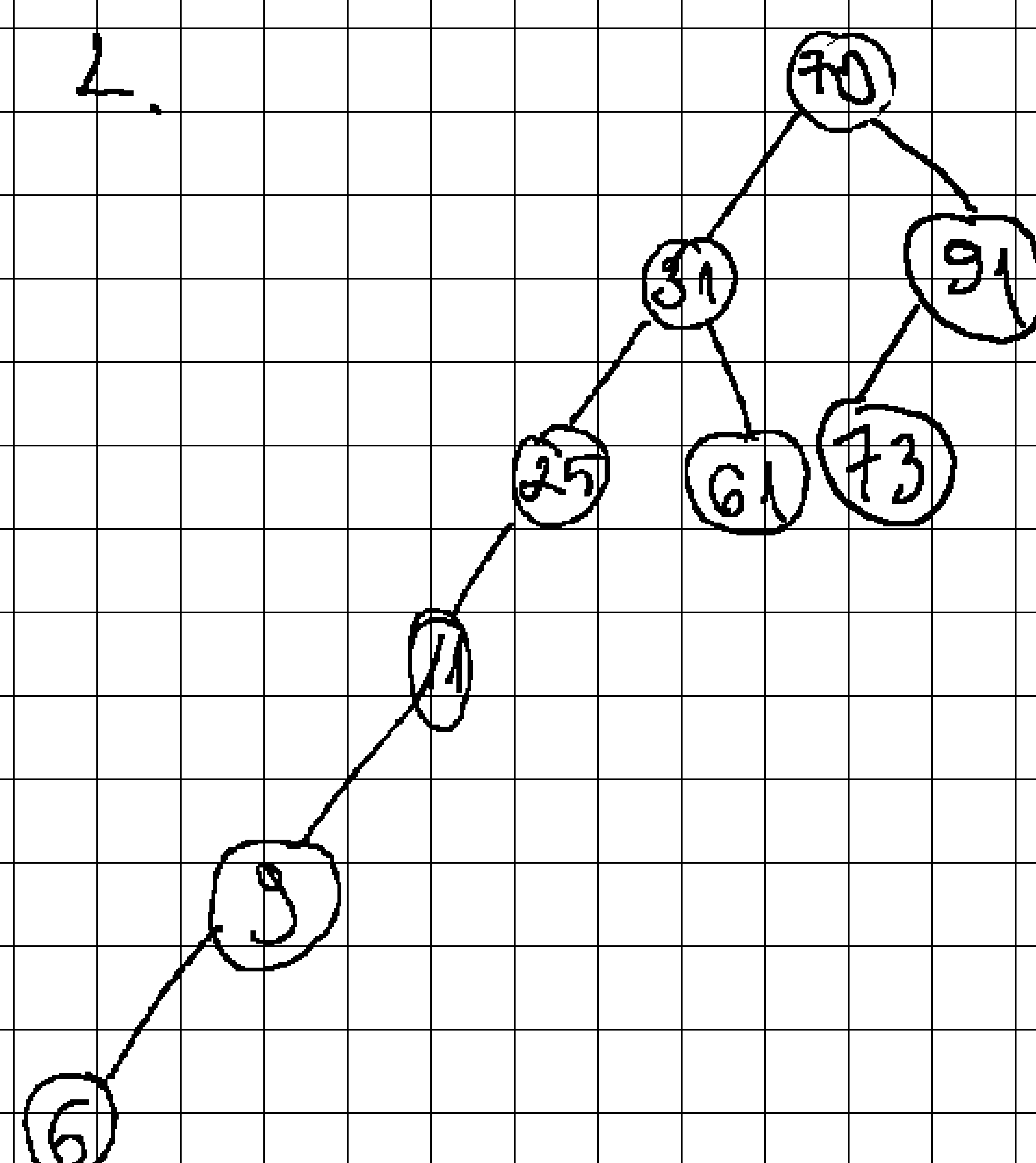
Delete 62:

1.



We replace 62 with the max of the left subtree

2.



We replace 62 with the min of the right subtree

3. $\boxed{a)}$ $N^2 + N^2 * \log_2 N \in \Omega(N^2)$ True

$$\lim_{N\to\infty} \frac{N^2+N^2 * \log_2 N}{N^2} = \lim_{N\to\infty} \frac{\cancel{N^2}}{\cancel{N^2}} + \lim_{N\to\infty} \frac{\cancel{N^2} * \log_2 N}{\cancel{N^2}} = 1+\infty = \infty$$

b) $N^2 + N^2 * \log_2 N \notin O(N^2)$

$$\lim_{n\to\infty} = \infty$$

c) $\lim_{n\to\infty} = \infty \quad \notin \Theta(N^2)$

$\boxed{d)}$ $N^2 + N^2 * \log_2 N \in O(N^3)$ True

$$\lim_{n\to\infty} \frac{N^2+N^2 * \log_2 N}{N^3} = \underbrace{\lim_{n\to\infty} \frac{1}{N}}_{0} + \underbrace{\lim_{n\to\infty} \frac{\log_2 N}{N}}_{0} = 0$$

as N approaches infinity faster

2. a. pop - $\Theta(1)$ because we only need to access the last element and pop it.

b. push when the number of elems is less then the capacity $\Rightarrow$ the resize does not occur ($O(n)$) and the push is $\Theta(1)$ as we just add the elem

c. isEmpty is $\Theta(1)$ as we only need to check if the curr. no. of elems is zero

d) top - $\Theta(1)$ because we only need access to the last elem in the stack (it does not depend on N)

$\boxed{e)}$ none of these operations

3. a. binary search tree:

WC: $O(N)$ for all operations when we have a degenerate tree: (example, inserting 1,2,3,4,5....)

b. Dynamic array

push - $\Theta(n)$ if we need to mantain the array sorted

pop - $O(n)$ if we have to remove an elem from the middle or if it's sorted

top - $O(n)$ - if it's unsorted $\Theta(1)$ - sorted

$\boxed{c.}$ binary heap

push $\Rightarrow$ $\Theta(\log_2 n)$ - insert at the end then do a bubble up which take $\log_2 n$ operations

pop $\Rightarrow$ $O(\log_2 n)$ - remove the root by replacing it with the last element in the heap then do a bubble down

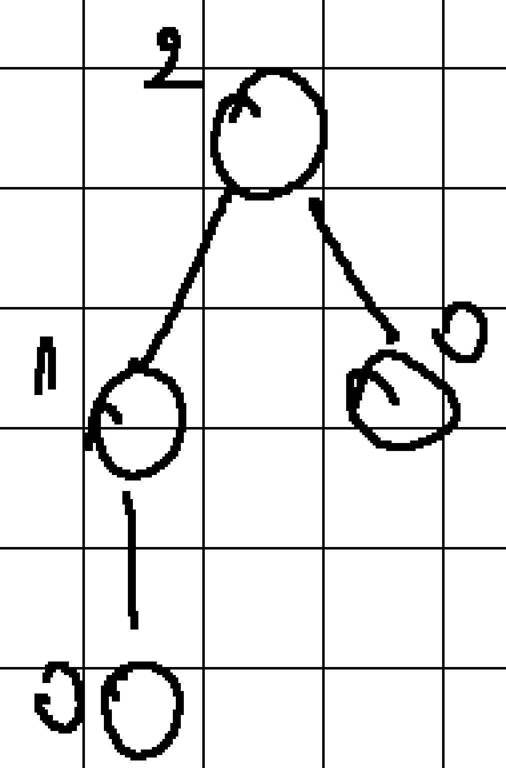
top $\Rightarrow$ $\Theta(1)$ $\Rightarrow$ the root is always the min or max, so we can find the elem. with the highest priority

d. AVL Tree

$O(\log_2 N)$ as we need to traverse the tree for each operation

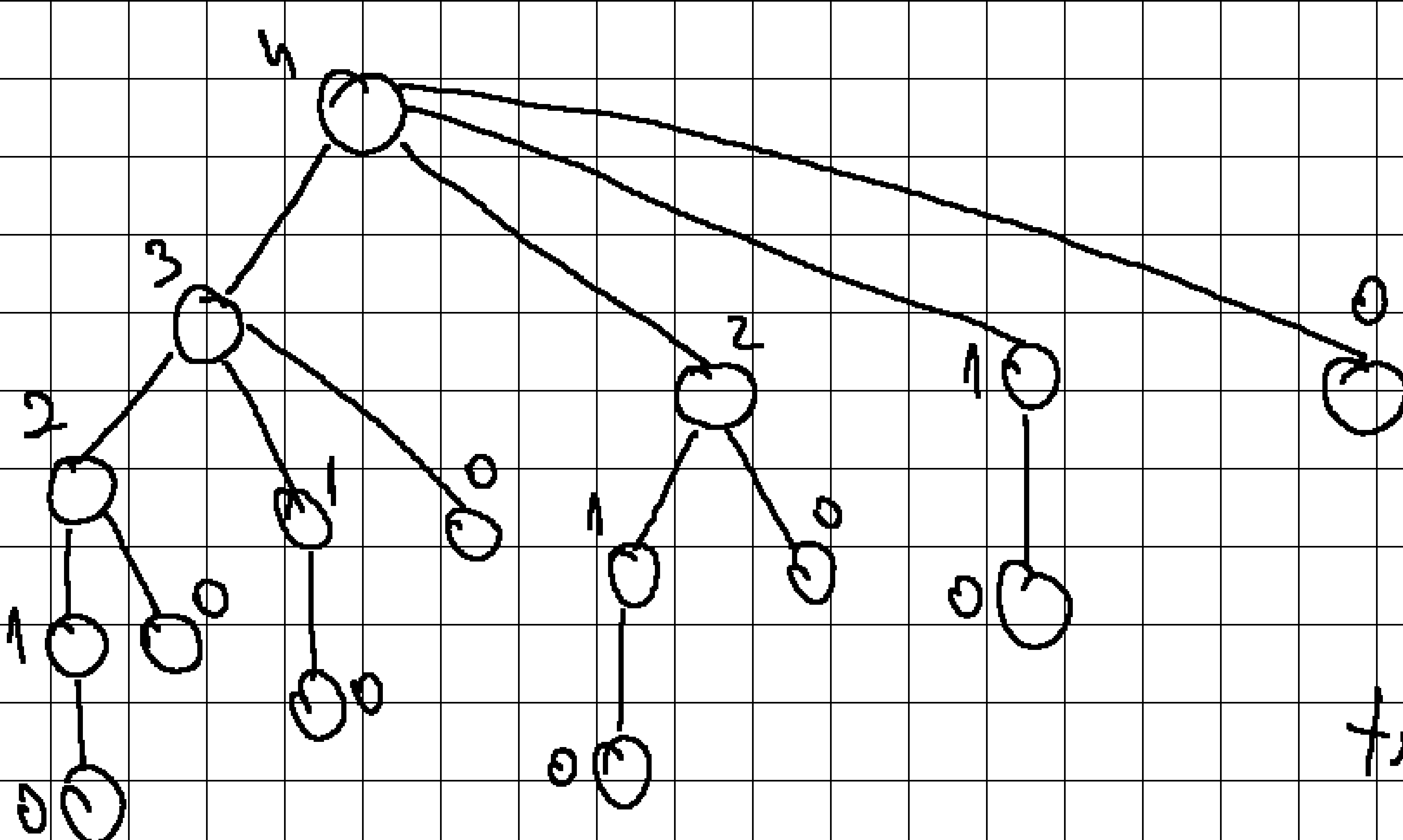
4. $7 = 0111_{(2)} \Rightarrow 2^0+2^1+2^2 = 7$

$\Rightarrow$ we have binomial trees of order 0,1,2



$24 = 2^4+2^3 = 11000_{(2)} \Rightarrow$

$\Rightarrow$ we have binomial trees of order 3,4



Merging:

$$\begin{array}{r} 11000+ \\ 00111 \\ \hline 11111 \end{array} = 31$$

we have trees of order 0,1,2,3,4

$\boxed{e.5}$

5. A queue is a FIFO

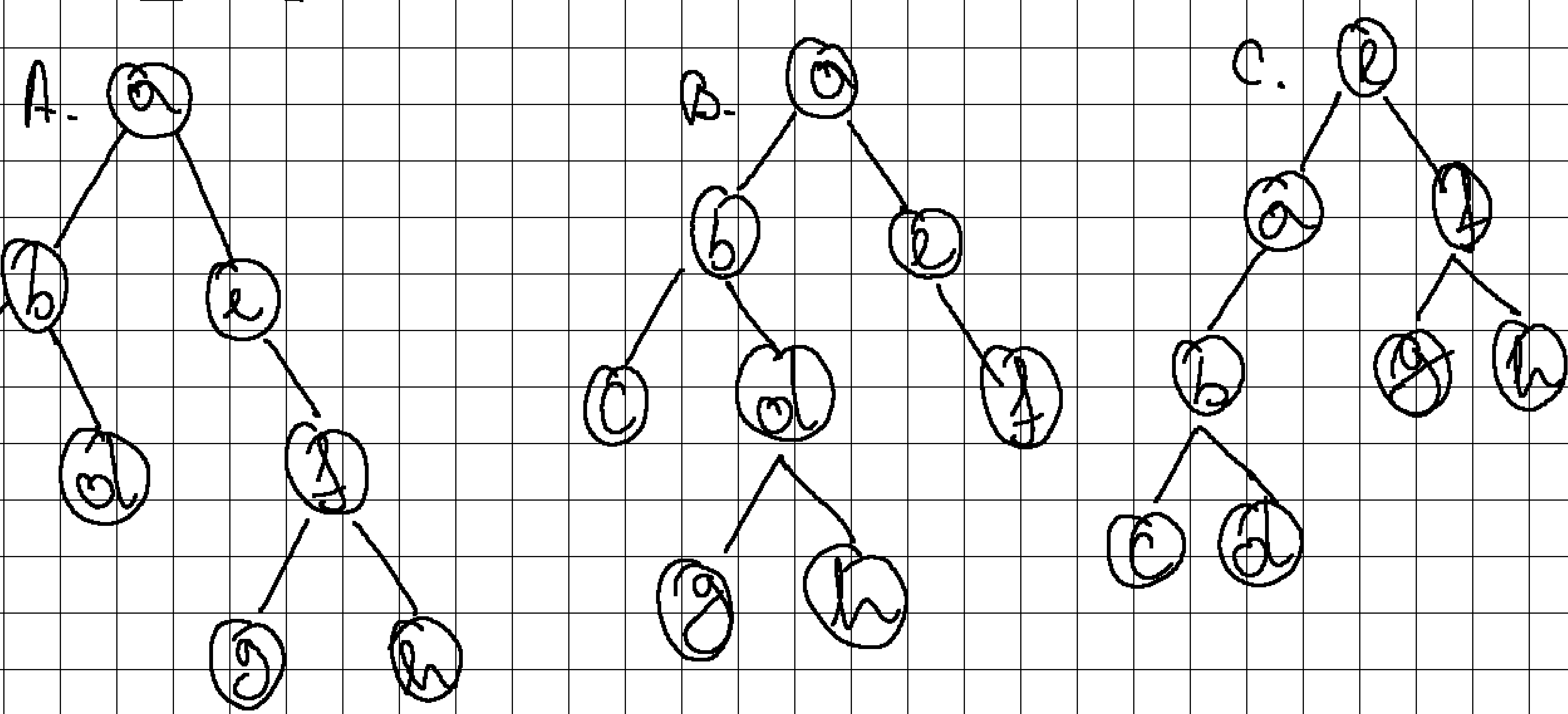
$\Rightarrow$ we need operations to add elements to the back and remove from the front

d. push_back and pop_front

6. c) we need to know all the elements that will be inserted from the beginning

Exam Row 8:

2.



A: preorder: a b c d e f g h
inorder: c b d a e g f h
postorder: c d b g h f e a
levelorder: a b e c d f g h

B: preorder: a b c d g h e f
inorder: c b g h d a e f
postorder: c g h d b f e a
levelorder: a b e c d f g h

C: preorder: e a b c d f g h
inorder: c b d a e g f h
postorder: c d b a g h f e
levelorder: e a f b g h c d

A,C - inorder
A,B - levelorder

b) $m = 11$

$h_1(x) = x \% m$

$h_2(x) = \text{sum\_digits}(x) \% m$

0	1	2	3	4	5	6	7	8	9	10
			25		71				9	

0	1	2	3	4	5	6	7	8	9	10

$h_1(71) = 71 \% 11 = 5$

$h_1(9) = 9 \% 11 = 9$

$h_1(25) = 25 \% 11 = 3$

$h_1(69) = 69 \% 11 = 3$

$h_2(25) = 7 \% 11 = 7$

$h_1(52) = 52 \% 11 = 8$

$h_1(8) = 8 \% 11 = 8$

$h_2(52) = 7 \% 11 = 7$

$h_1(25) = 25 \% 11 = 3$

$h_2(69) = 15 \% 11 = 4$

$h_1(16) = 16 \% 11 = 5$

$h_2(71) = 8 \% 11 = 8$

$h_1(41) = 41 \% 11 = 8$

$h_2(8) = 8 \% 11 = 8$

$h_1(71) = 5$

$h_2(16) = 7 \% 8 = 7$

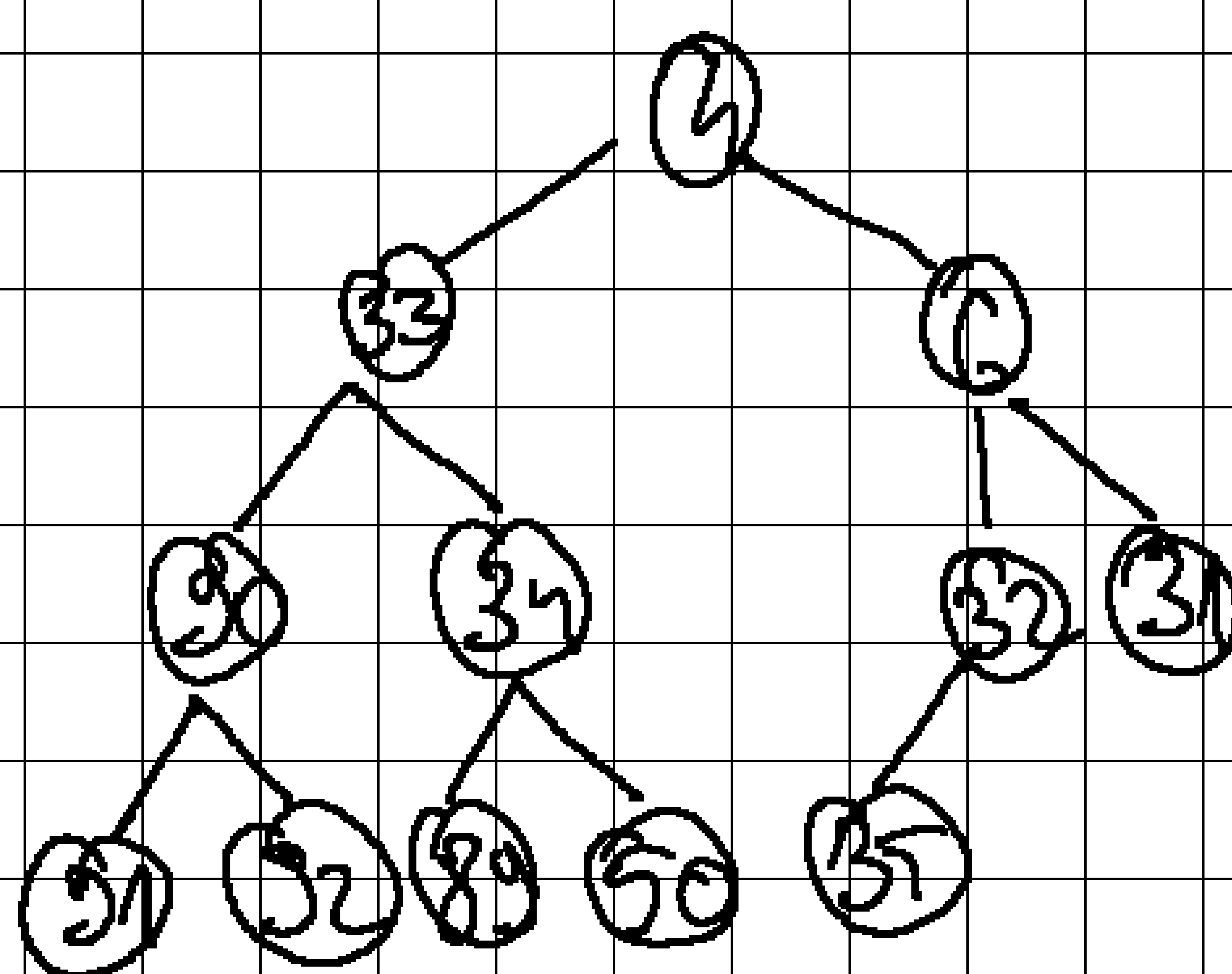
$h_1(52) = 8$

$h_2(41) = 5 \% 11 = 5$

0	1	2	3	4	5	6	7	8	9	10
			~~69~~ 25		~~71~~ ~~16~~ 71			~~52~~ ~~41~~ ~~8~~ 52	9	

0	1	2	3	4	5	6	7	8	9	10
				69	41		~~25~~ ~~52~~ 16	~~71~~ 8		

C.) [4, 33, 6, 90, 34, 32, 31, 91, 92, 89, 50, 35]

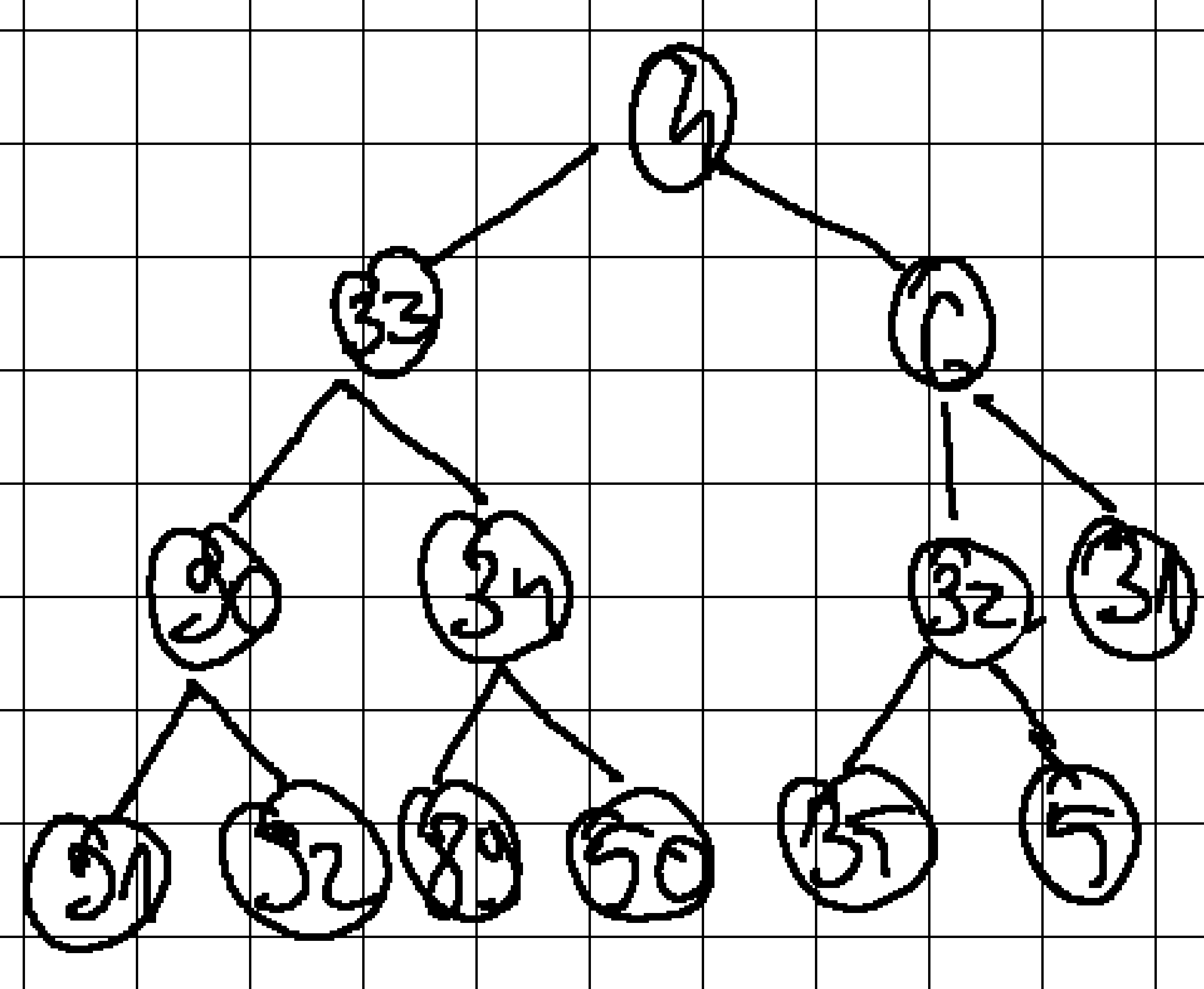


- heap structure holds (every node has 2 children and the two last levels are completed from left to right
- heap property is kept no need to swap elems:

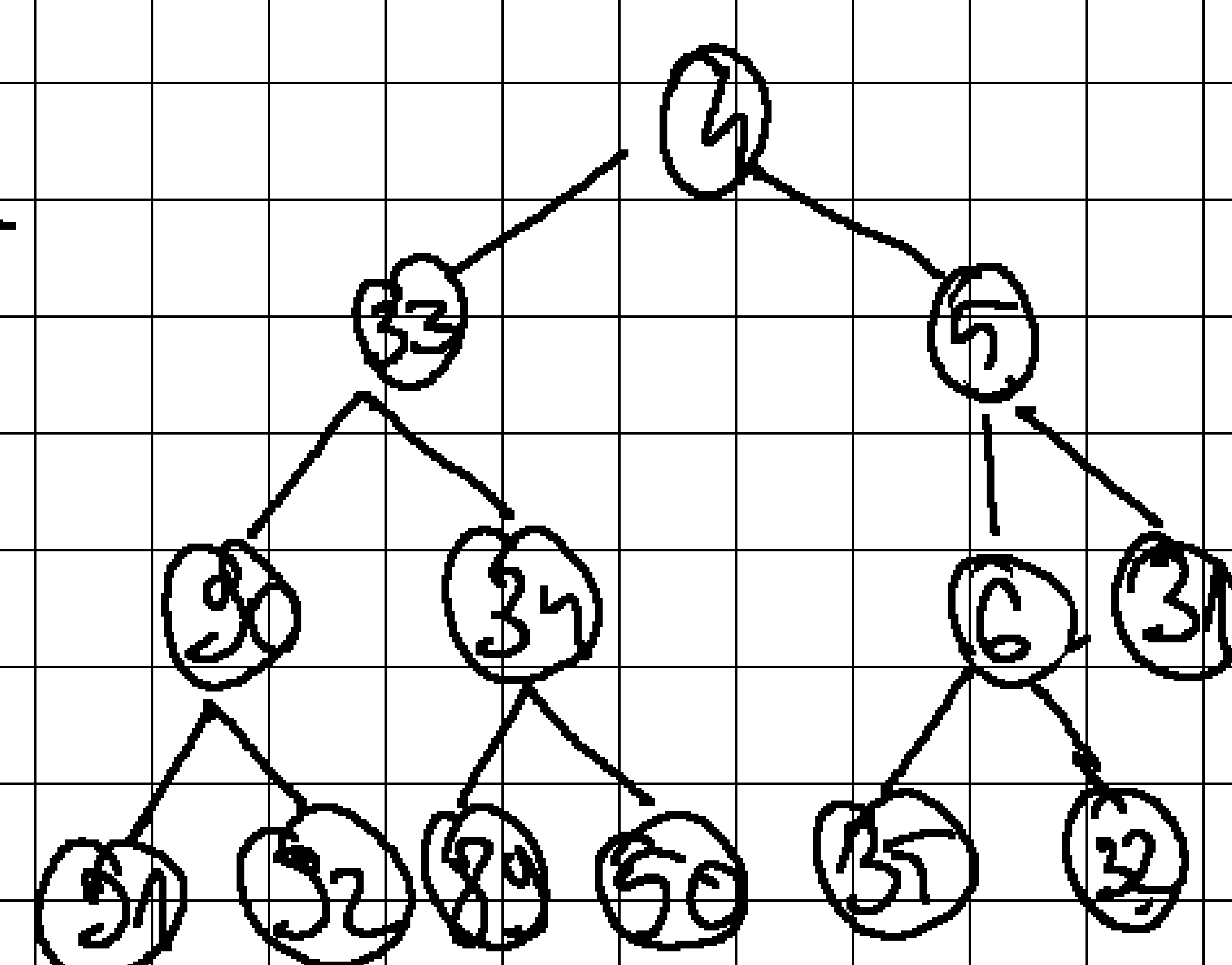
$a_i \le a_{2*i}$ or $a_i \le a_{2*i+1}$

$\Rightarrow$ min heap

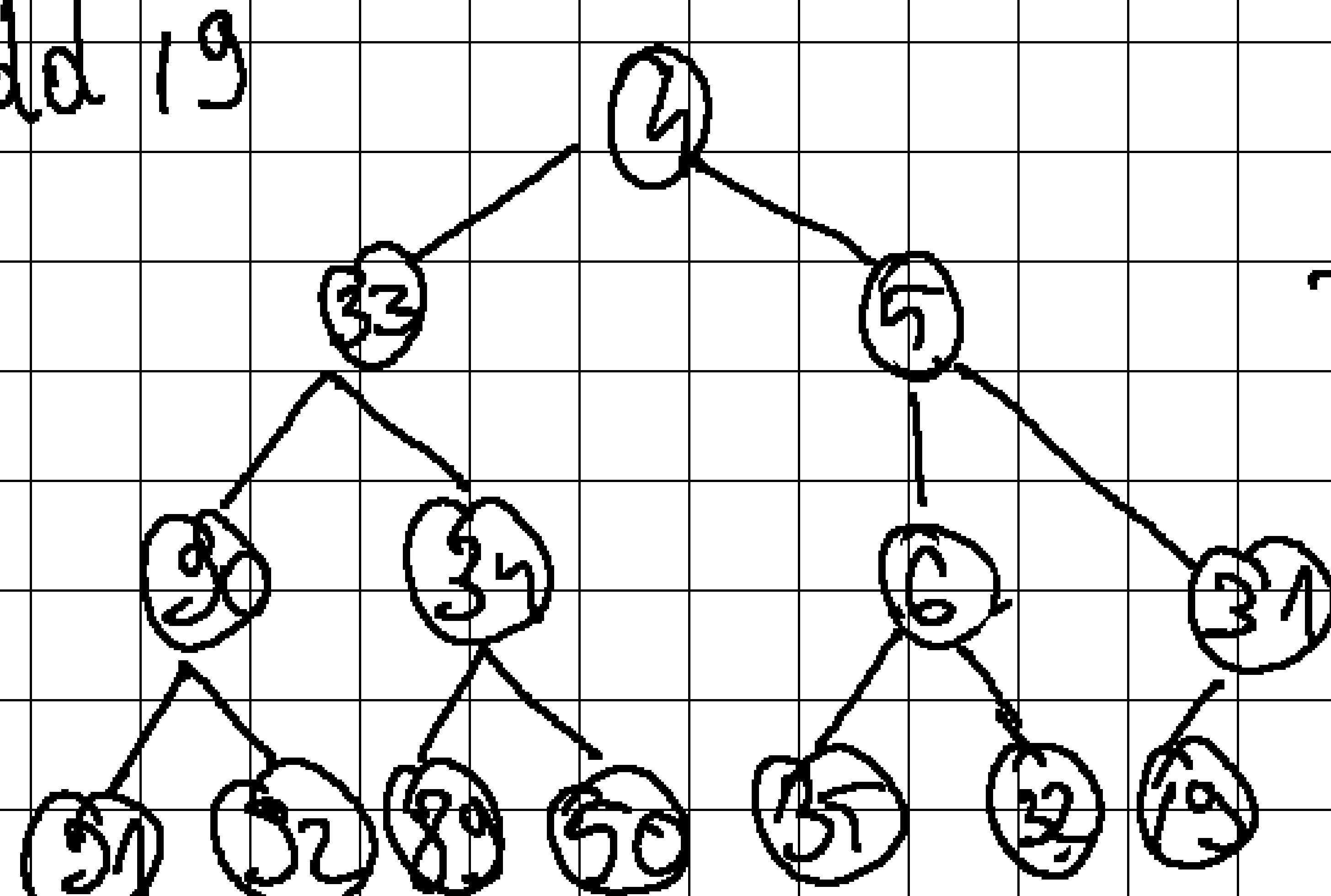
Add elem 5



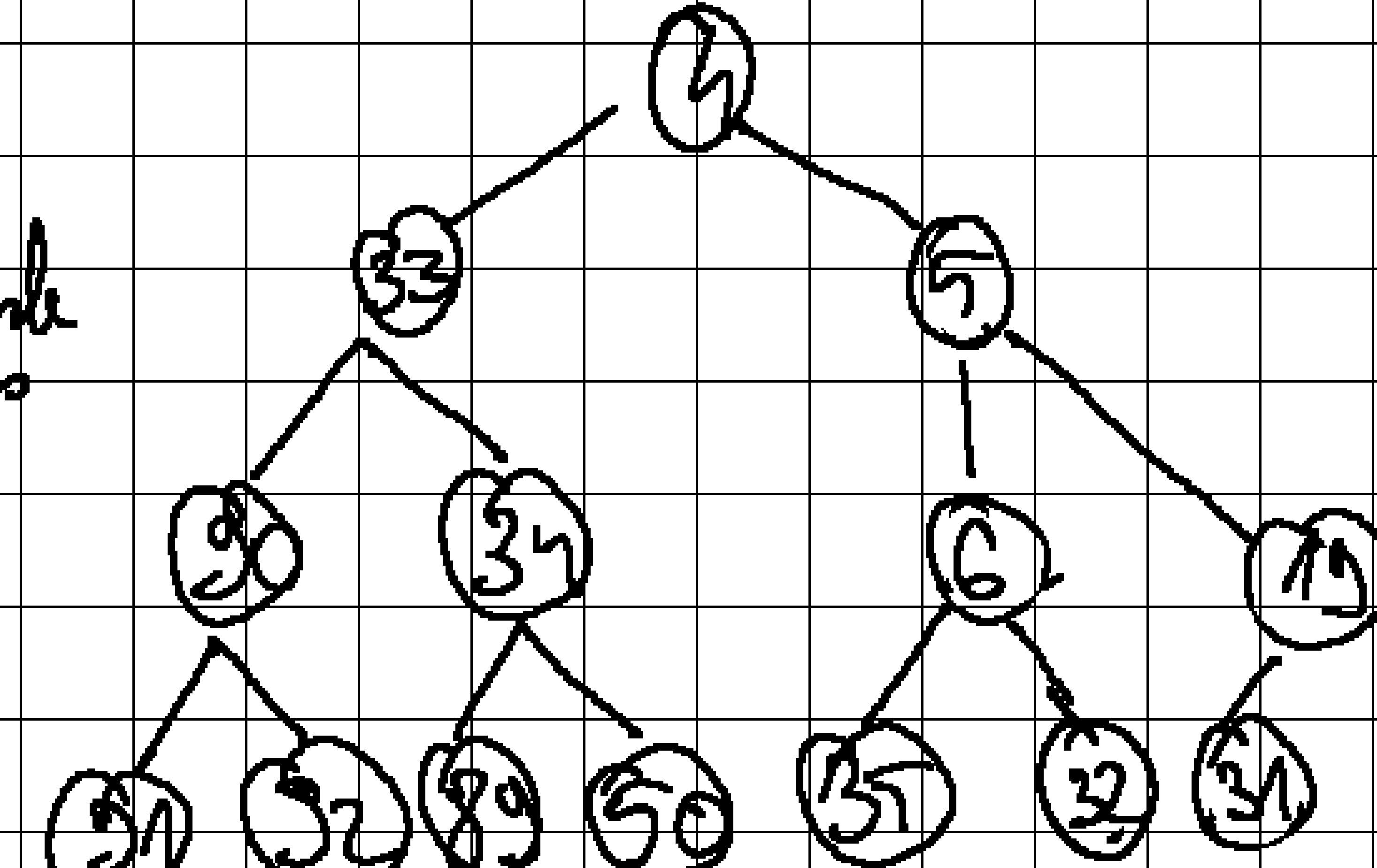
we need to do a bubble $\Rightarrow$ up



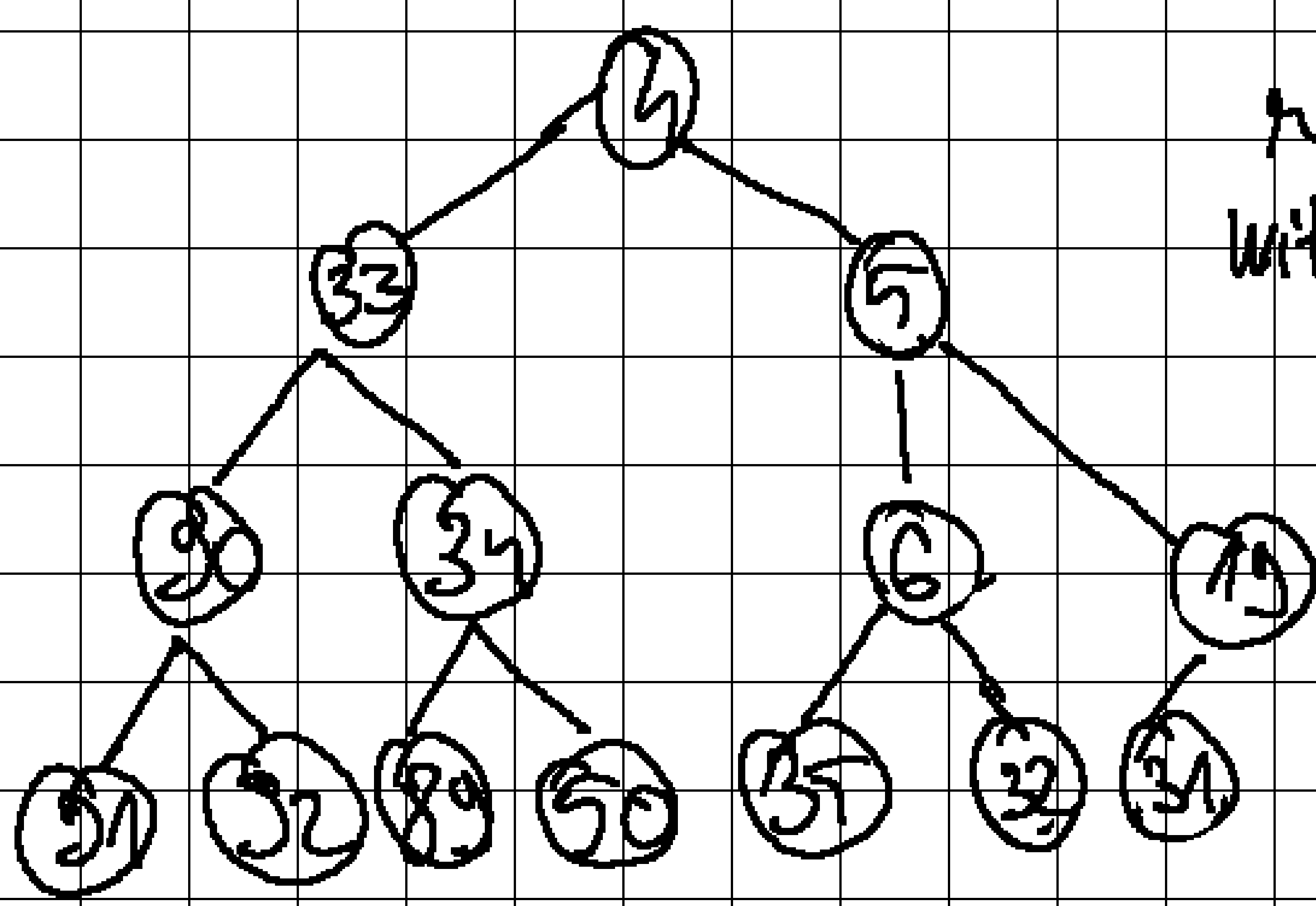
Add 19



we need to do a bubble $\Rightarrow$ up

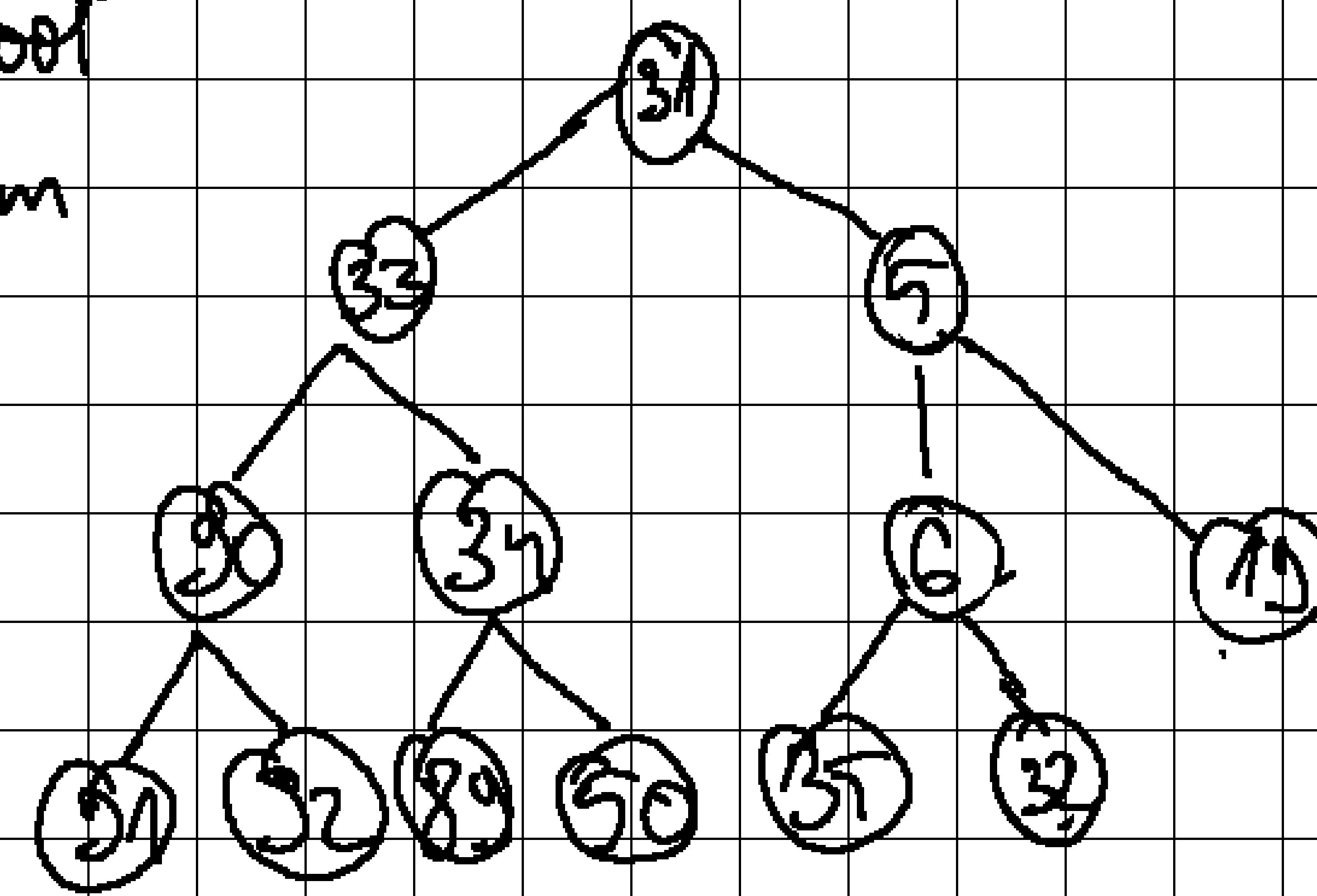


Remove one elem (we can only remove the root

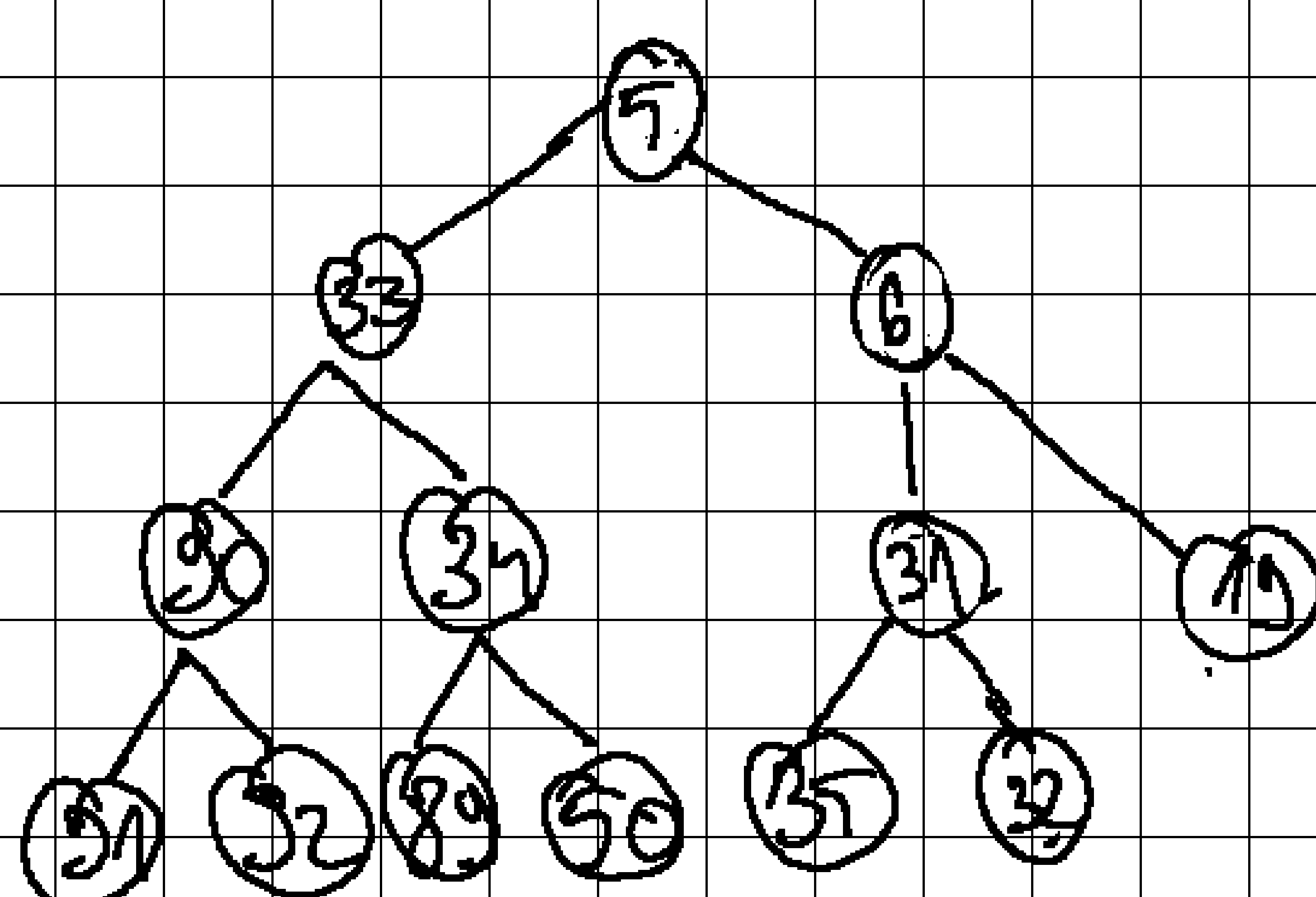


replace the root with the last elem

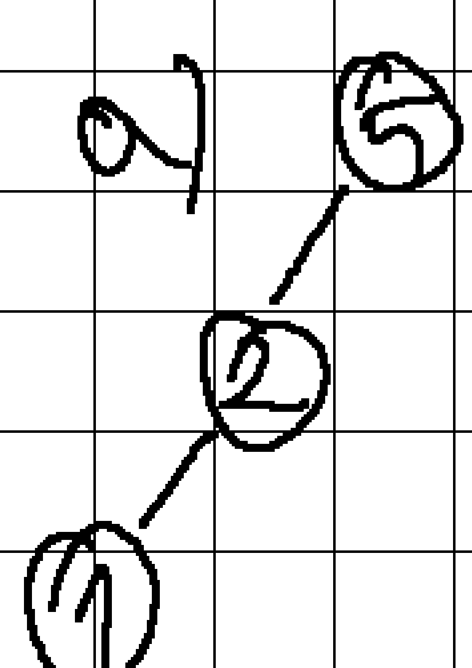
=>



We have to do a bubble down => to keep the heap property

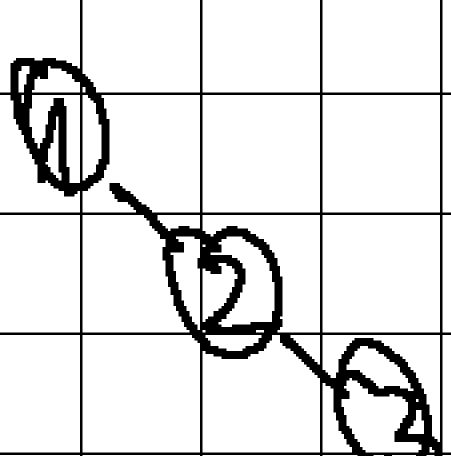


3. a)



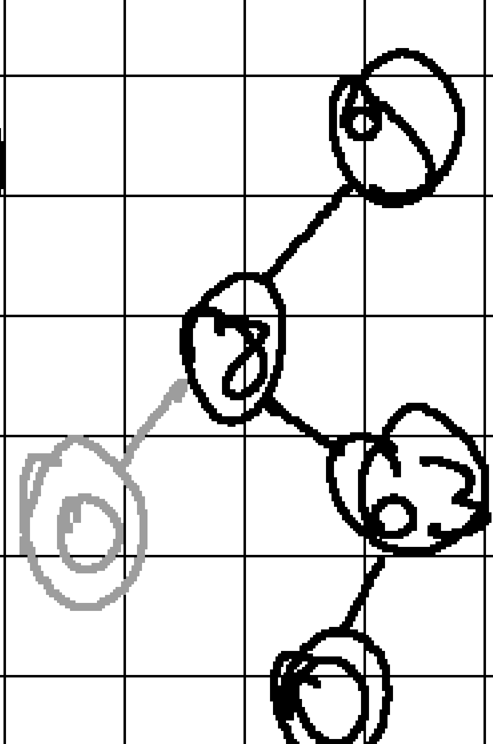
it should go to a value higher then 5 not less then it

b) 1,2,3,4 ...



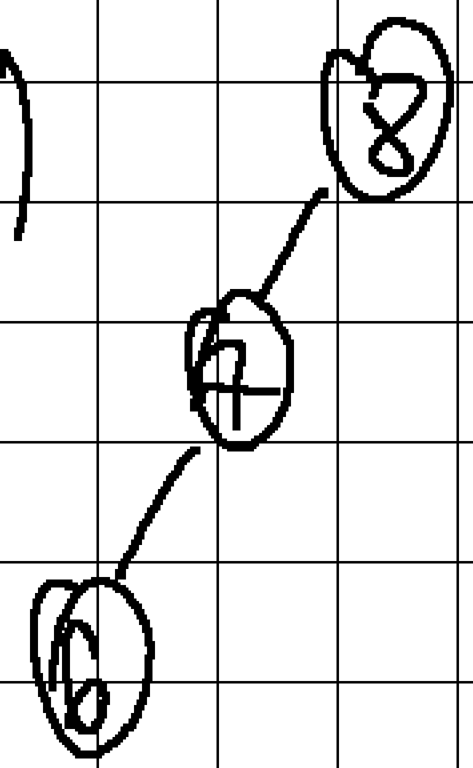
is just a linear search that stops at 8 which does not guarantee it finds 45

c)



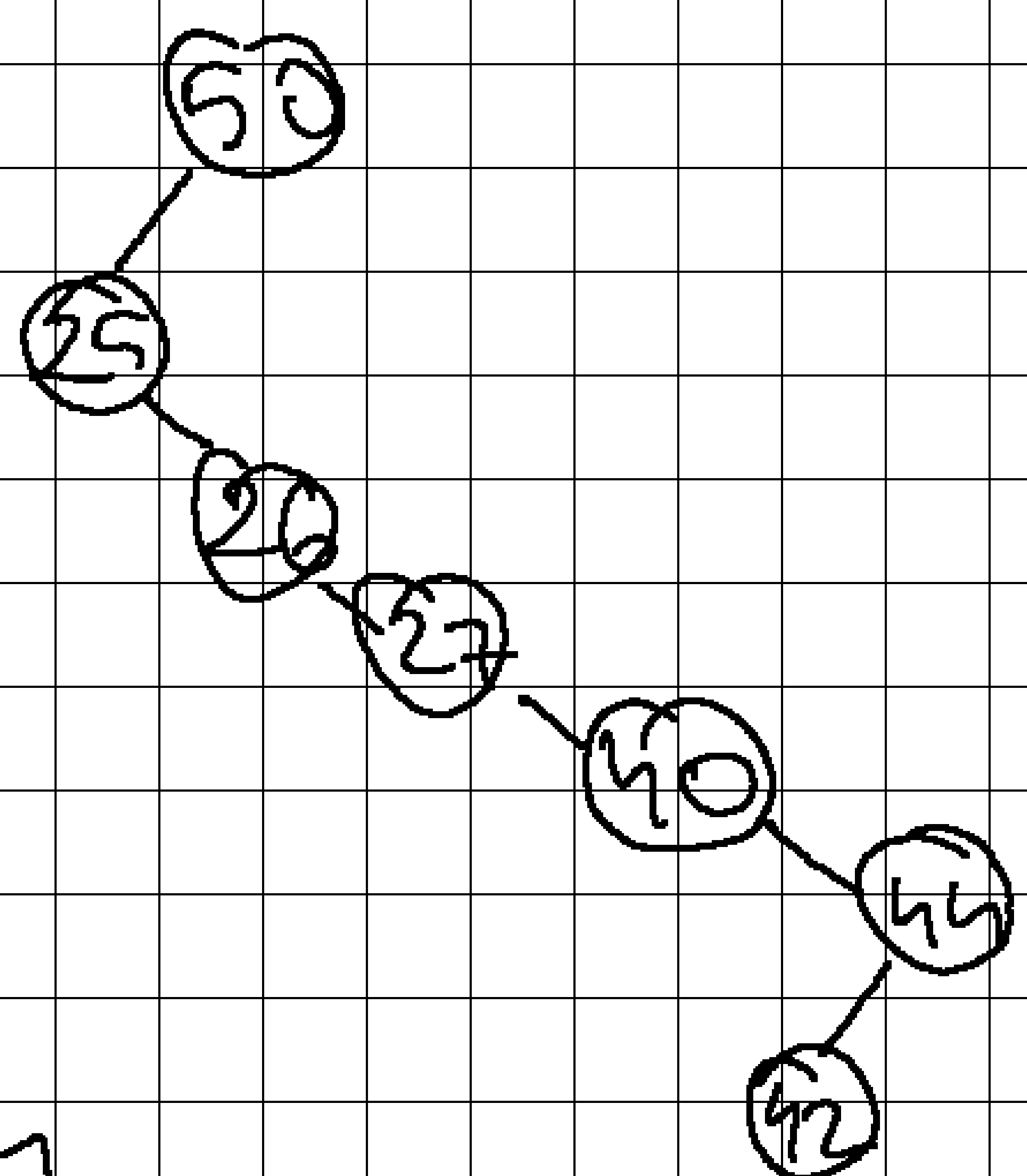
wrong , 0 cant be higher then 8

d)



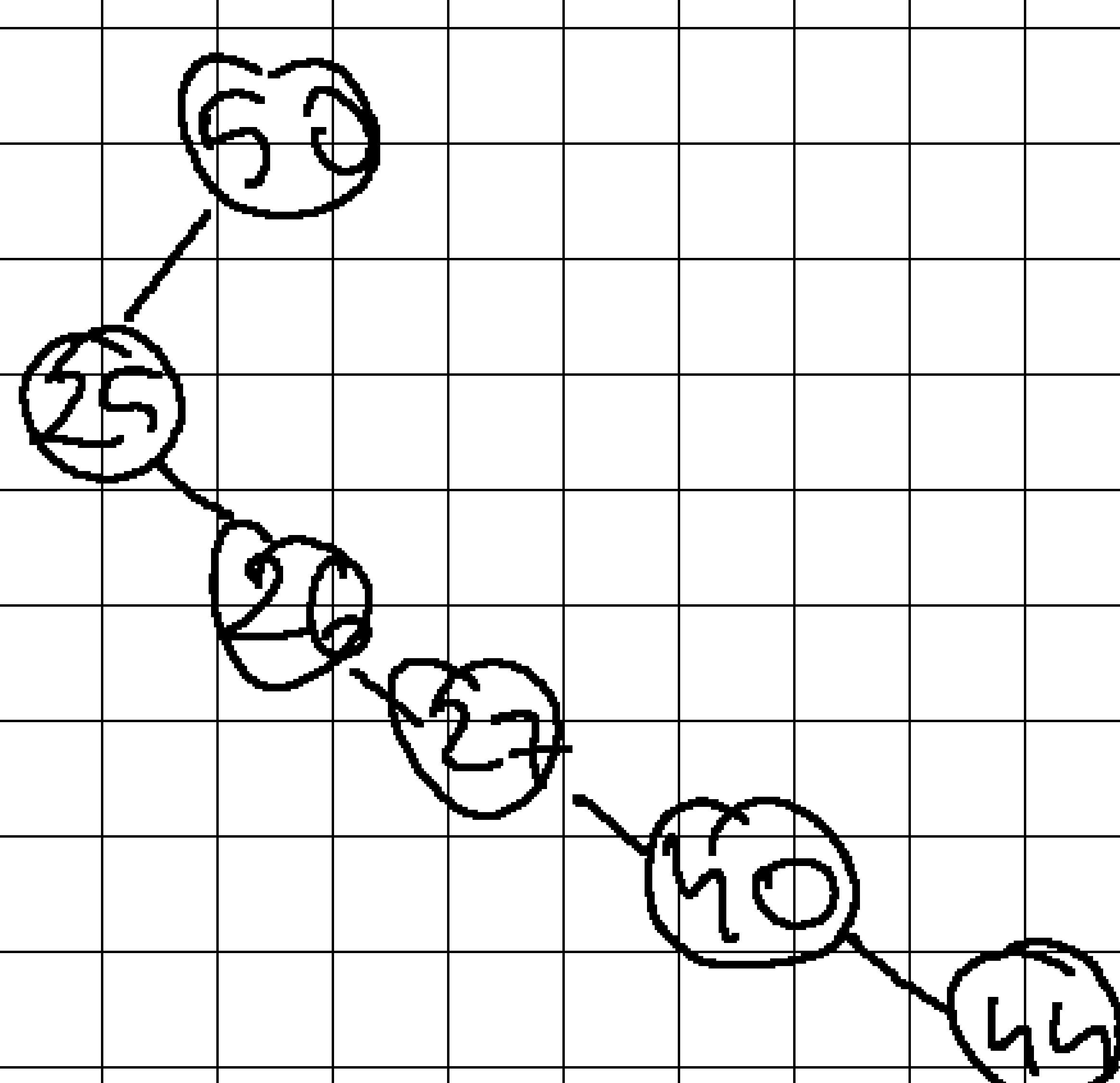
it keeps searching for an element lower then 8, which is wrong $45 > 8$

e)



It should have stopped at 44

$\boxed{f)}$



correct, it can find or not the value 45

2. $15n^3 + 1099n^2 + 5n\log_2 n = X$

a) $15n^3 + 1099n^2 + 5n\log_2 n \in \Theta(n^2)$ False

$$\lim_{n\to\infty} \frac{15n^3 + 1099n^2 + 5n\log_2 n}{n^2} = \lim_{n\to\infty} 15n + \lim_{n\to\infty} 1099 + \lim_{n\to\infty} \frac{5\log_2 n}{n} = \infty$$

$\boxed{b)}$ $X \in \Omega(n^2)$ True

$$\lim_{n\to\infty} X = \infty$$

c) $X \in O(n^2)$ False

$$\lim_{n\to\infty} X = \infty$$

$\boxed{d)}$ $X \in \Theta(n^3)$ True

$$\lim_{n\to\infty} \frac{15n^3 + 1099n^2 + 5n\log_2 n}{n^3} = \lim_{n\to\infty} 15 + \lim_{n\to\infty} \frac{1099}{n} + \lim_{n\to\infty} \underbrace{\frac{5\log_2 n}{n^2}}_{\to 0} = 15$$

e) $X \in O(n\log_2 n)$ False

$$\lim_{n\to\infty} \frac{15n^3 + 1099n^2 + 5n\log_2 n}{n\log_2 n} \lim_{n\to\infty} \underbrace{\frac{15n^2}{\log_2 n}}_{\to\infty} + \lim_{n\to\infty} \underbrace{\frac{1099n}{\log_2 n}}_{\to\infty} + \lim_{n\to\infty} 5 = \infty$$

4. 6 1 + 8 2 4 + * -

stack [~~6~~, ~~1~~, ~~7~~, ~~8~~, ~~2~~, ~~4~~, ~~6~~, ~~48~~

1+6 = 7

4+2 = 6

6*8 = 48

48 - 7 = 41

Result is 41

e) value between 15 and 100

5. d) push back pop front

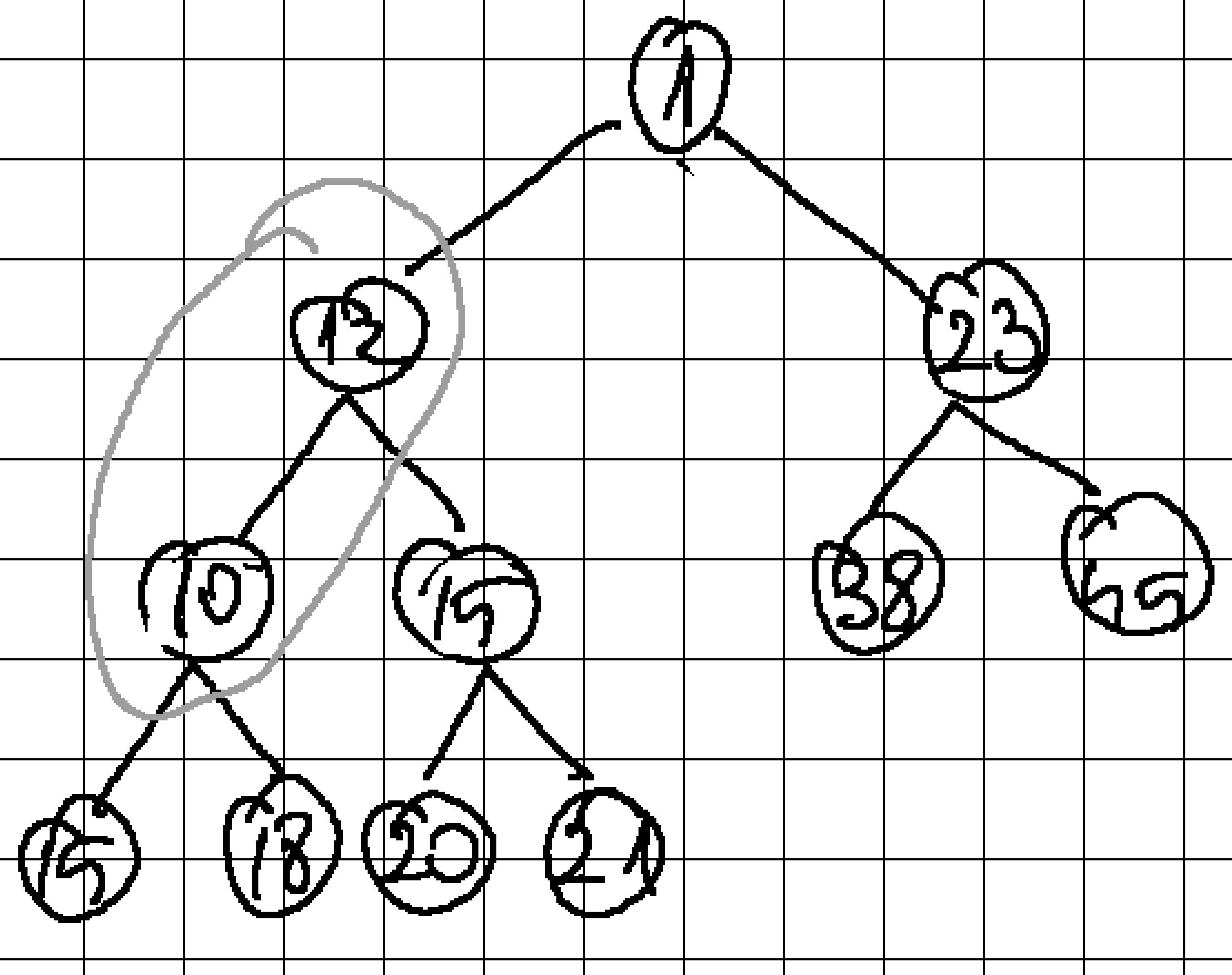
6. $19 = 2^4 + 2^1 + 2^0 = 10011$ => trees of order 4, 1, 0

=> there are 3 trees in the binomial heap

$\boxed{\text{C. 3}}$

Exam - row 6:

2. $A = [1, 12, 23, 10, 15, 38, 45, 15, 18, 20, 21]$

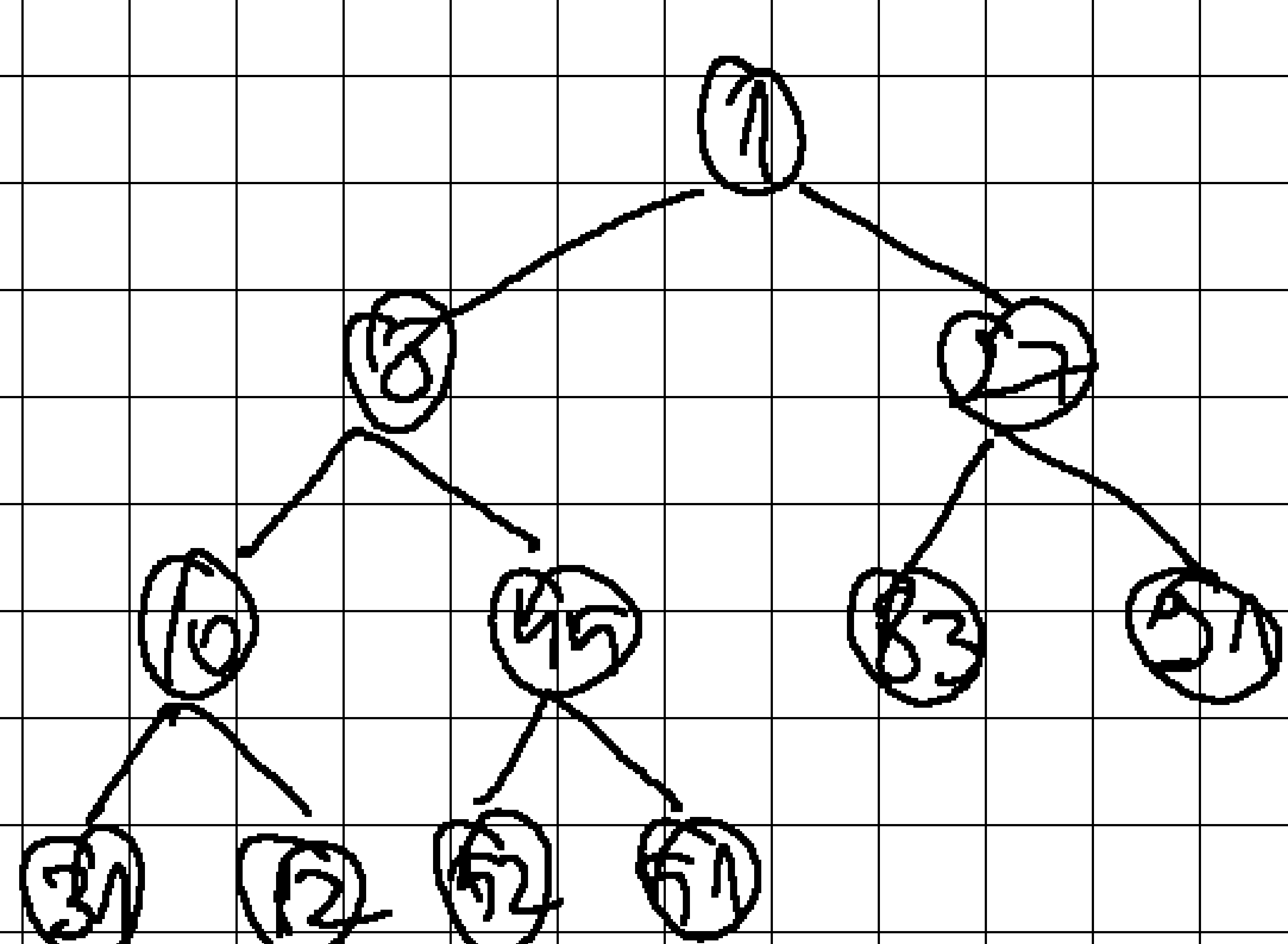


heap structure holds
every node has 2 children and on the last 2 levels we complete the tree from left to right
heap property does not hold

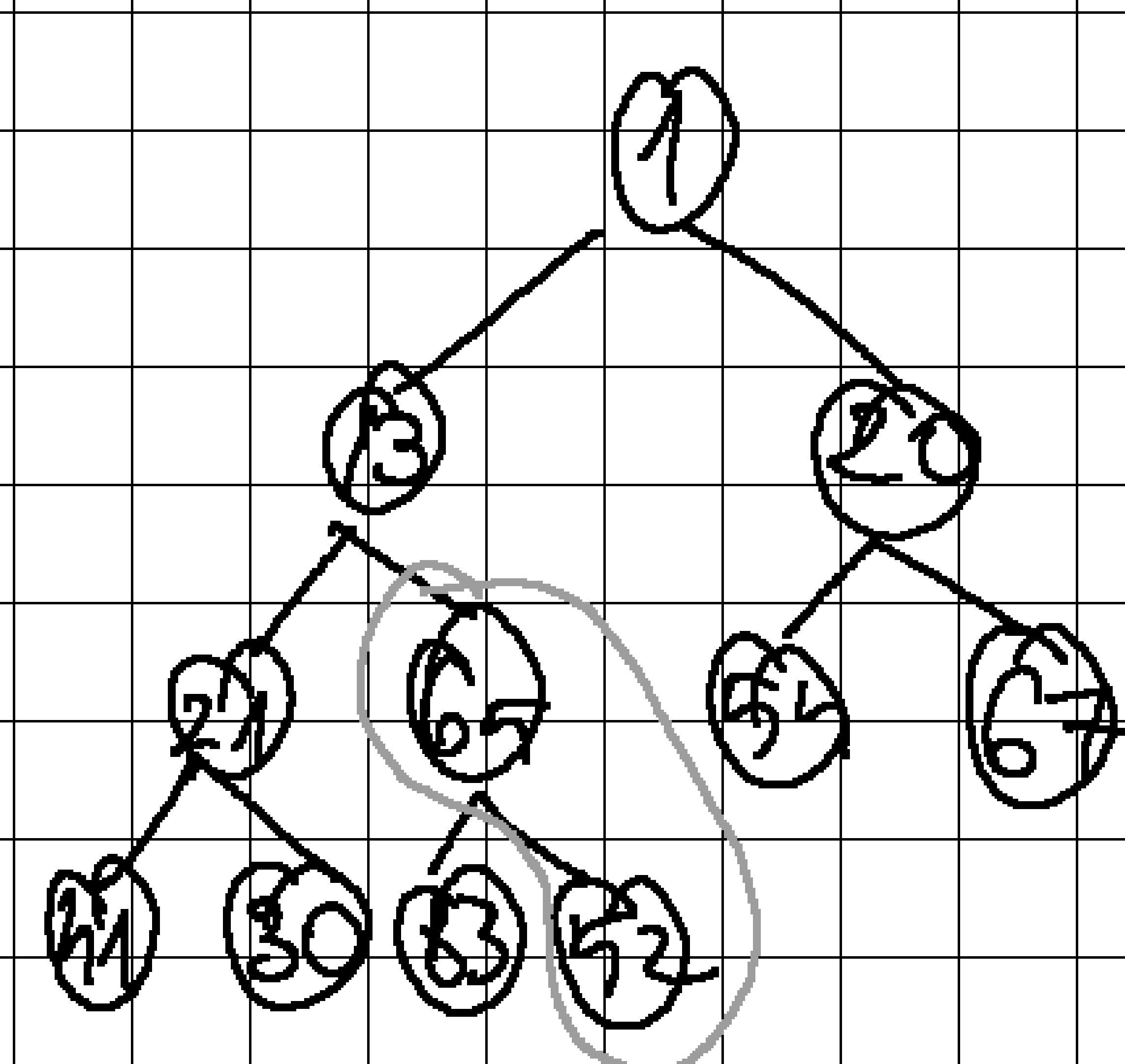
$$a_i \leq a_{2*i}$$
$$a_i \leq a_{2*i+1}$$

where the relation $\leq$ can be generalized

$B = [1, 8, 27, 10, 45, 83, 91, 31, 12, 52, 51]$



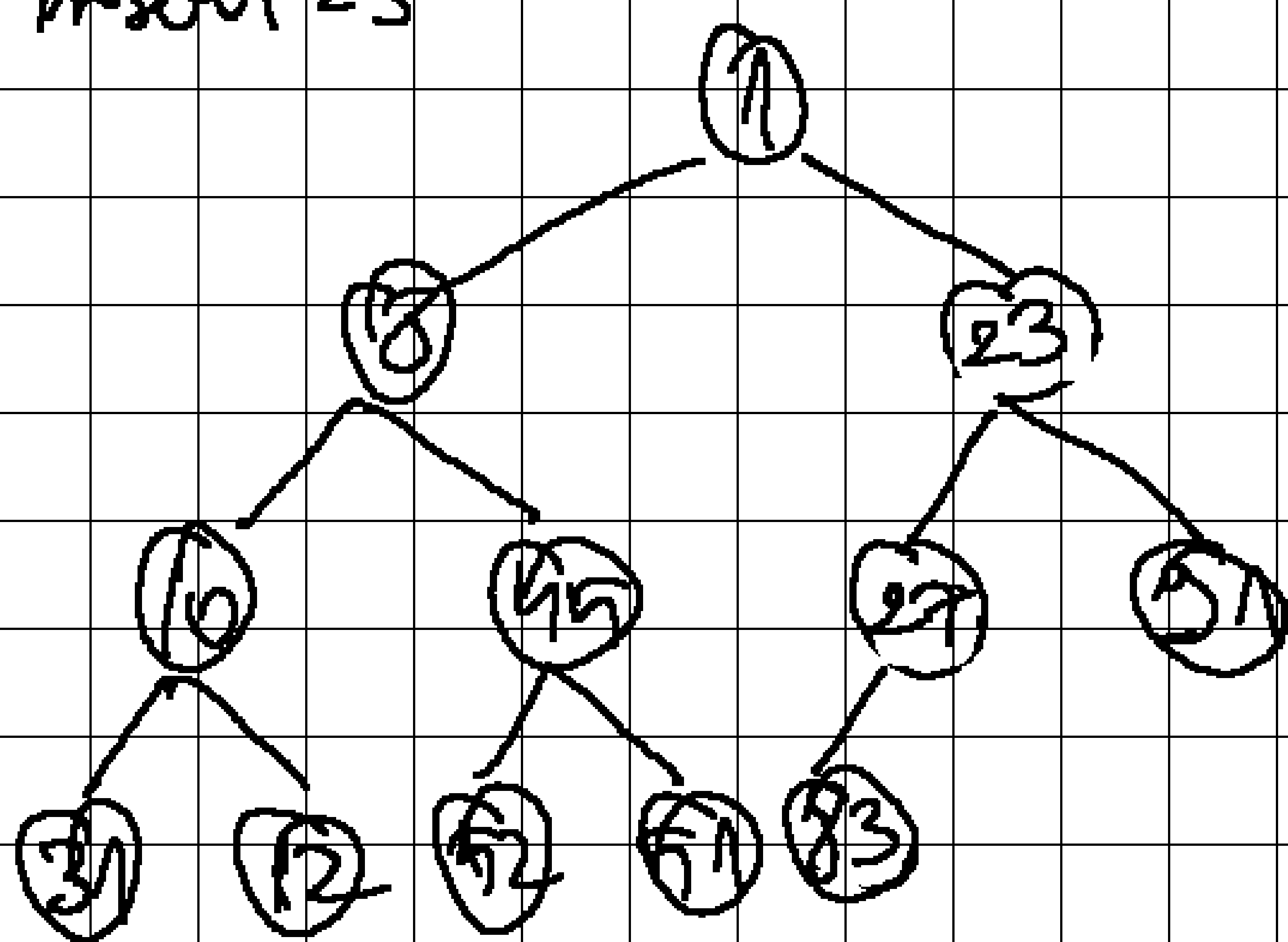
heap structure holds
heap property holds
we have a min heap



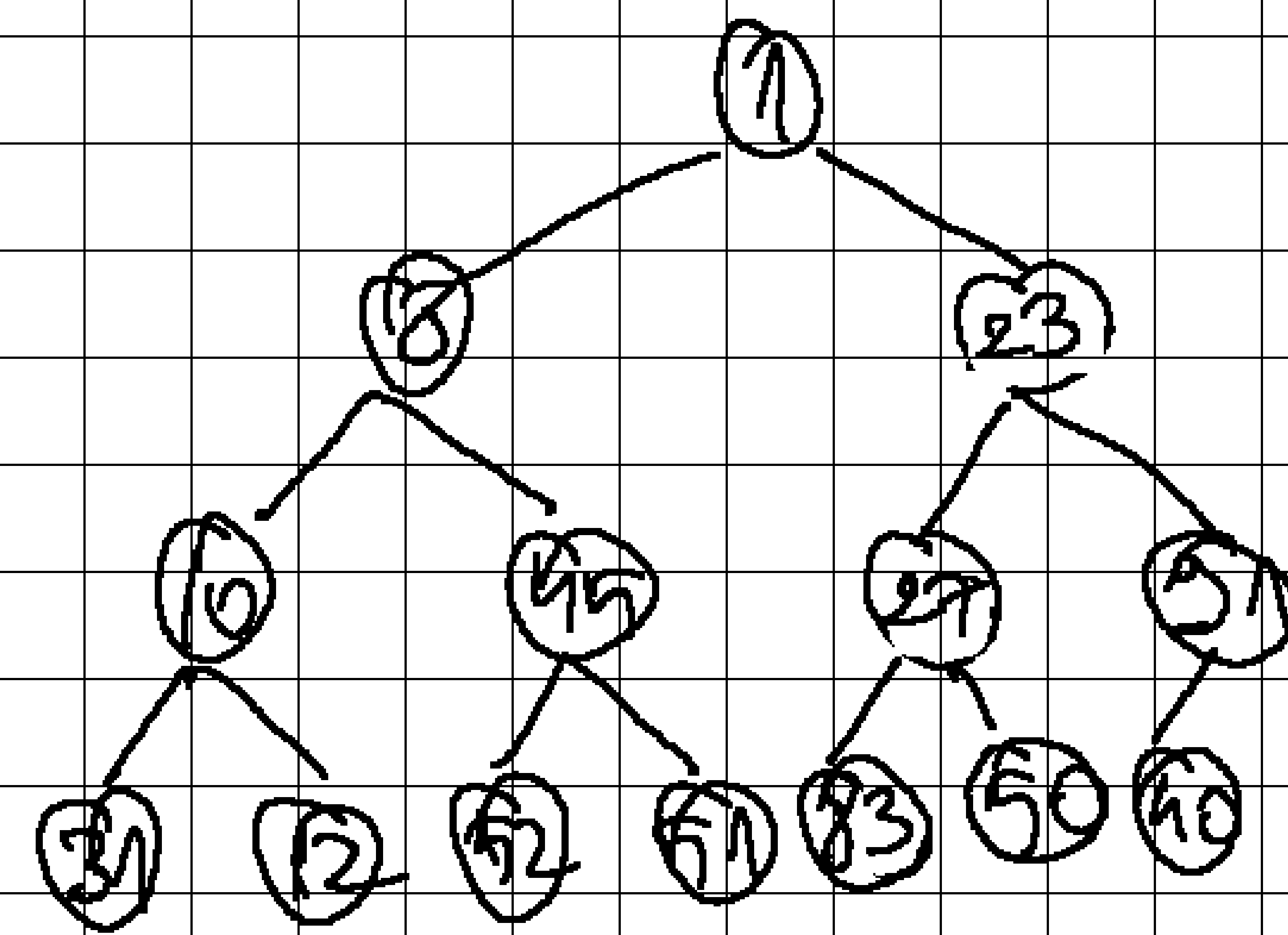
$C = [1, 13, 20, 21, 65, 54, 67, 41, 30, 83, 52]$

heap structure holds
heap property does not hold

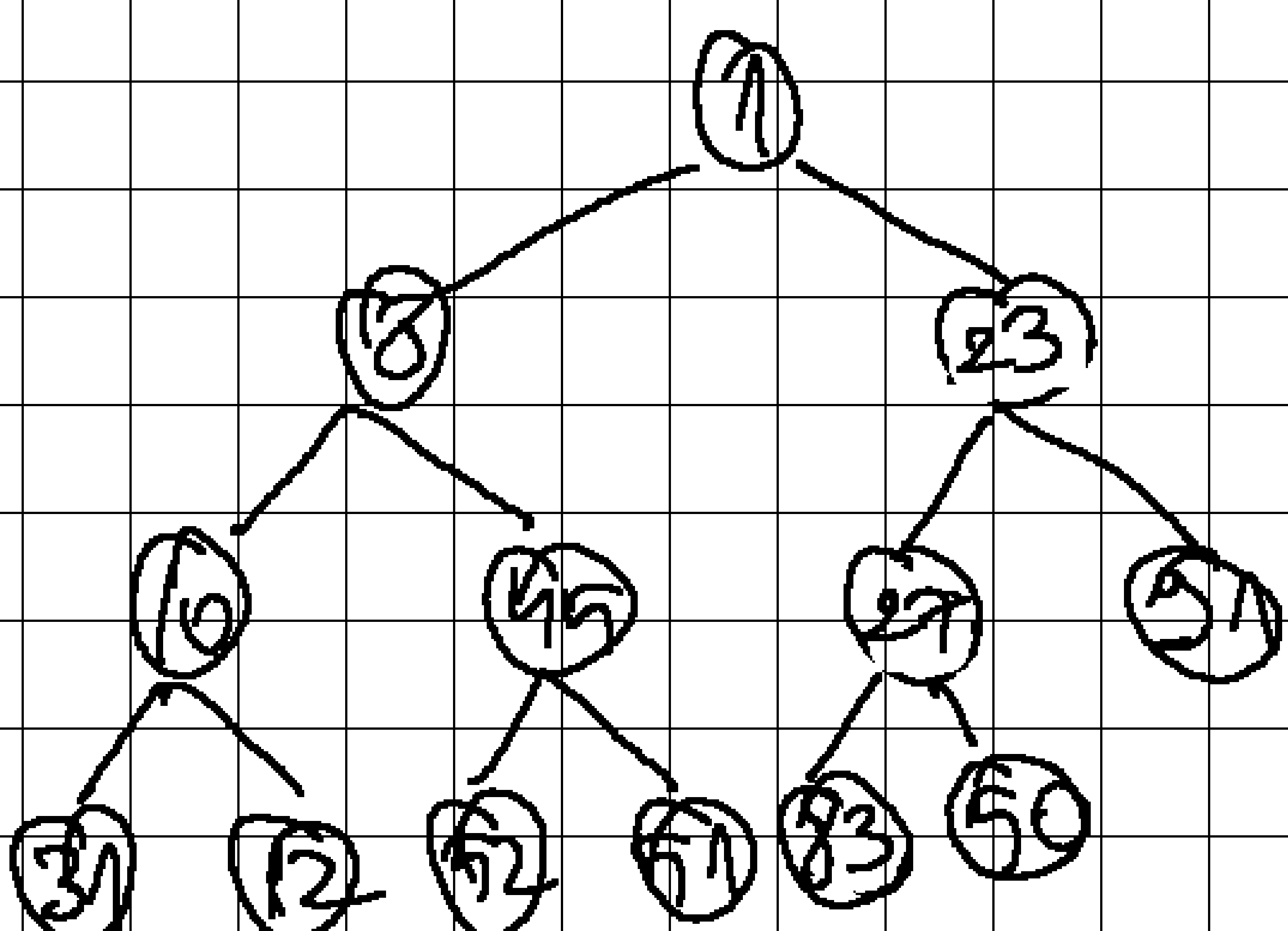
insert 23



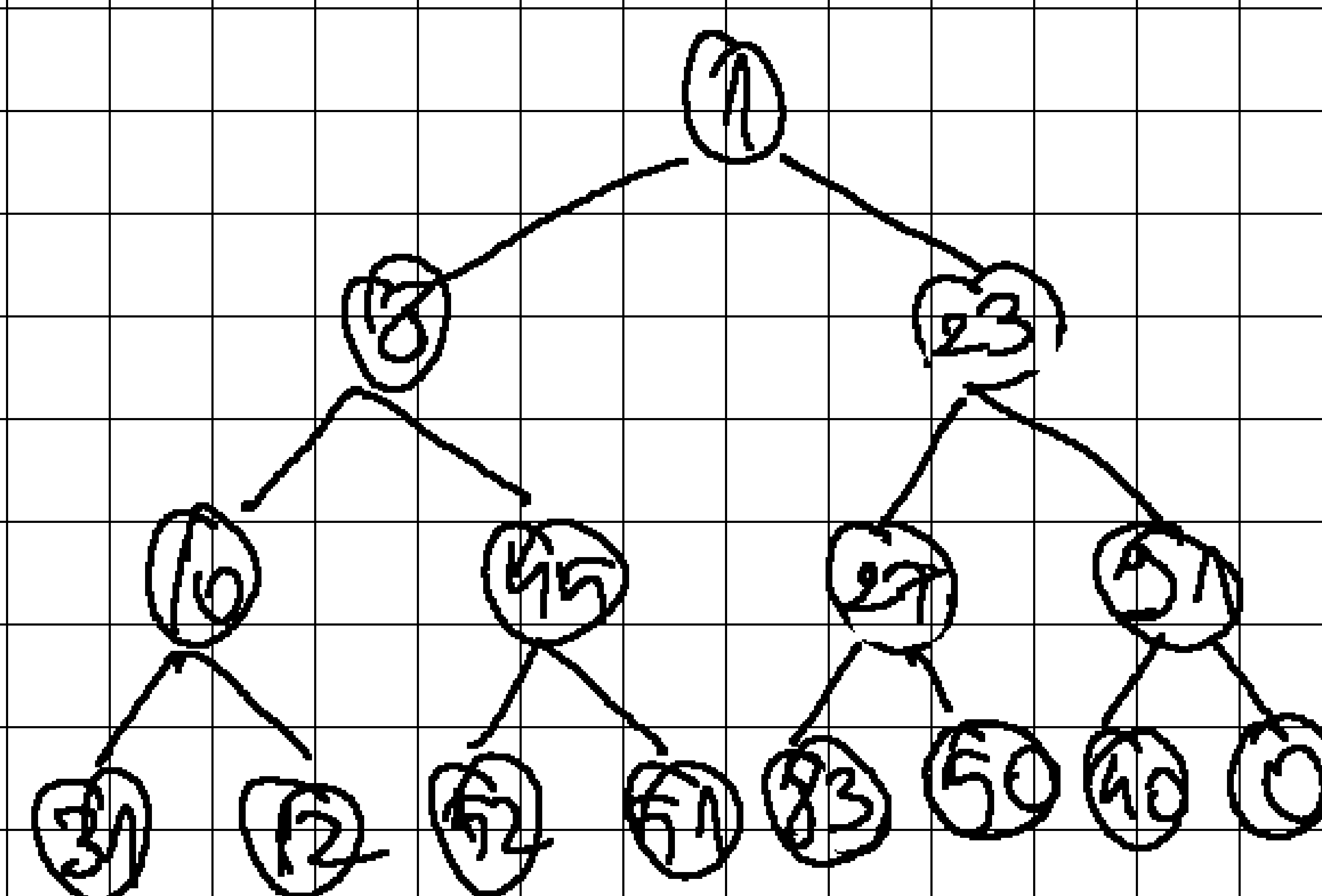
insert 40



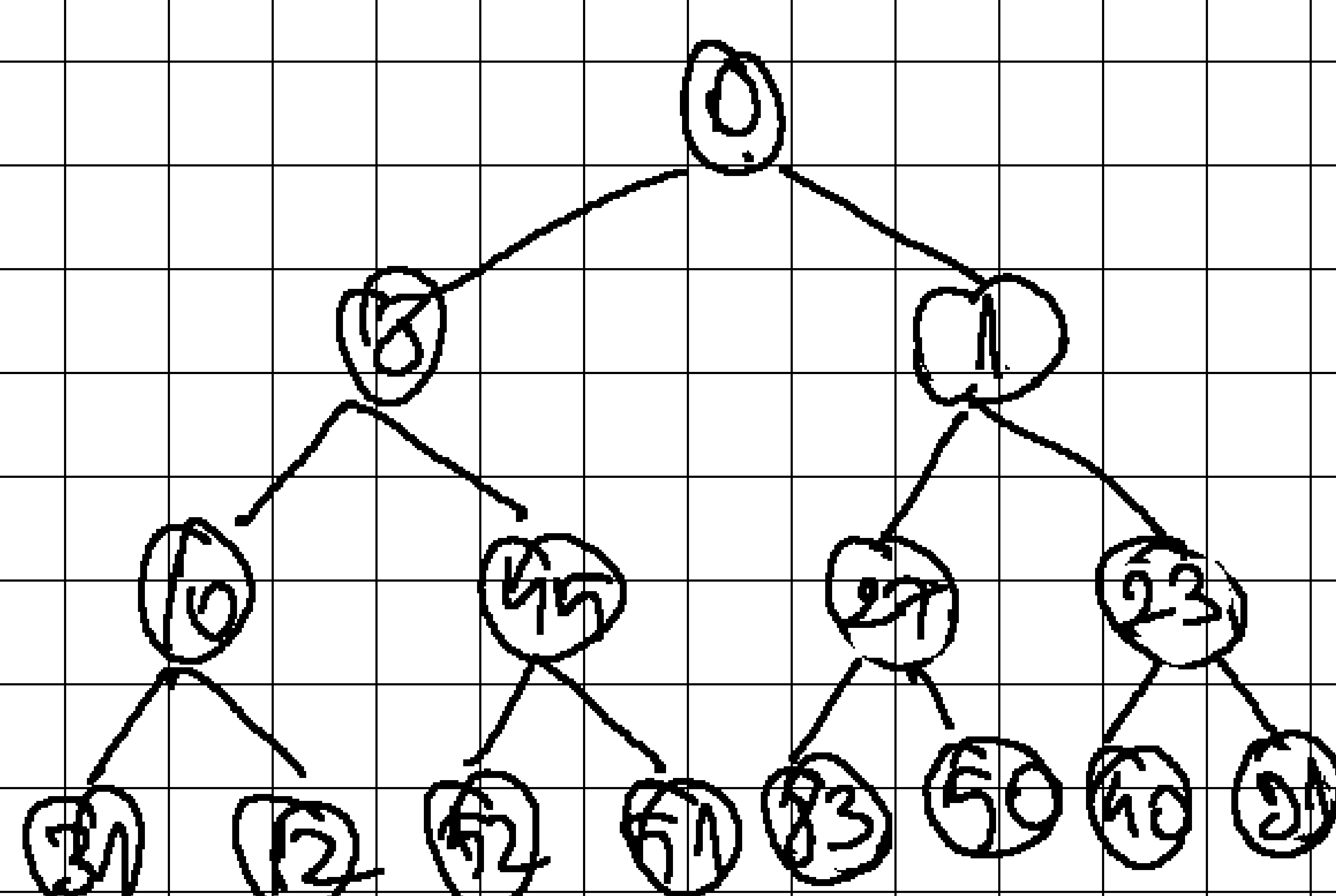
Insert 50



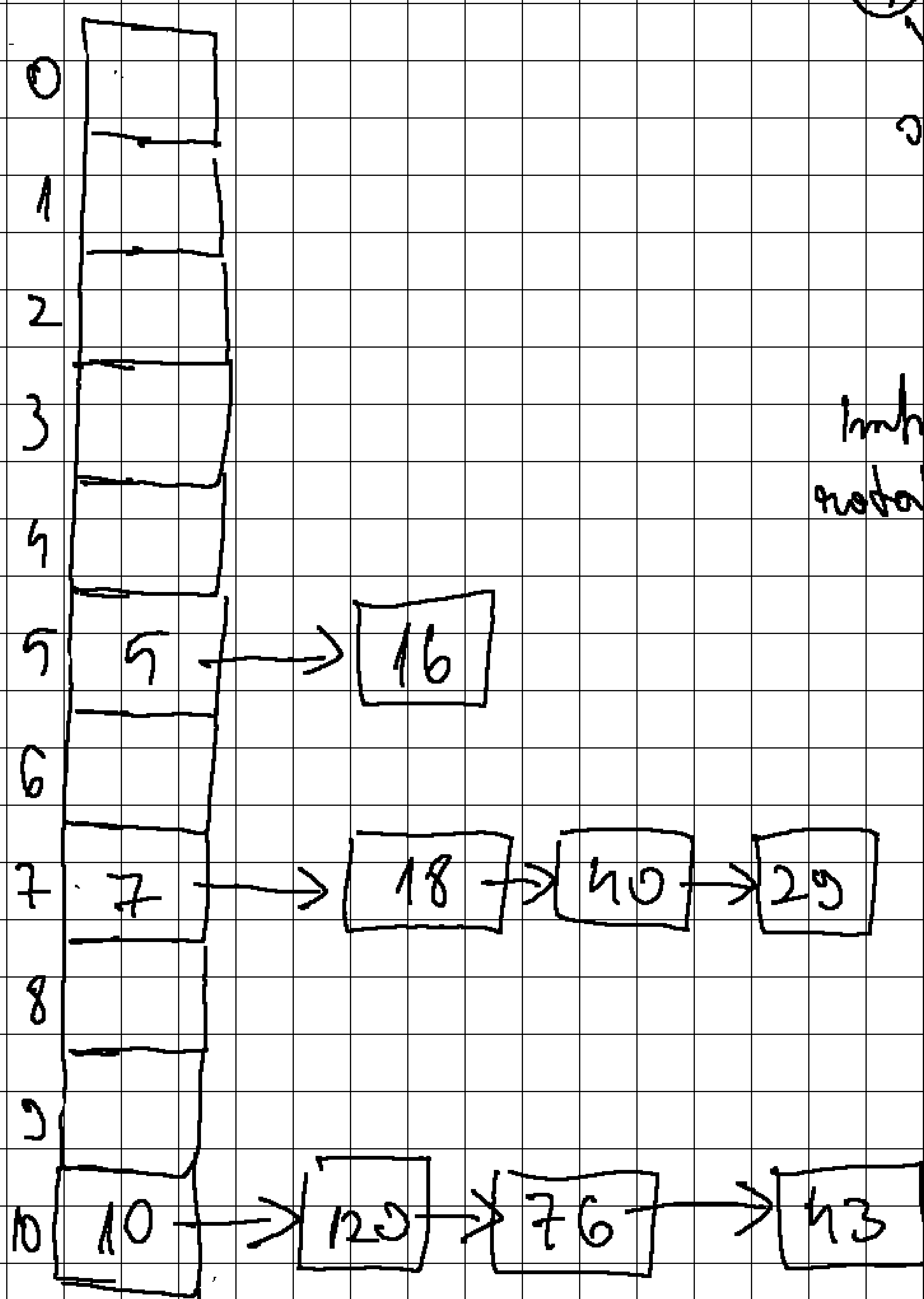
insert 0



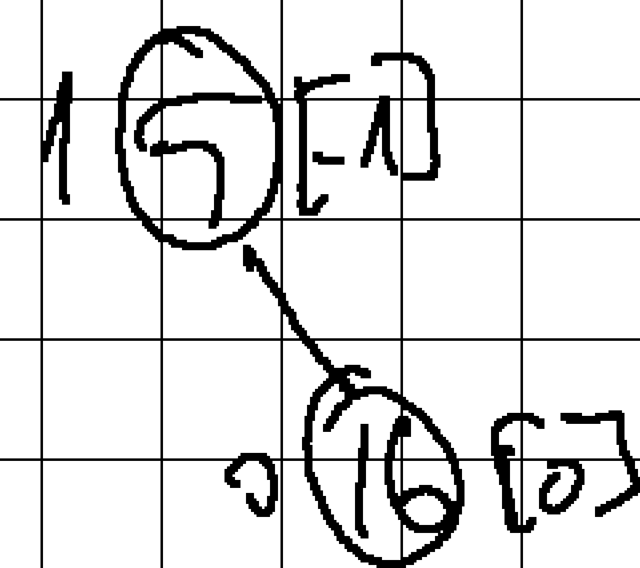
⇒



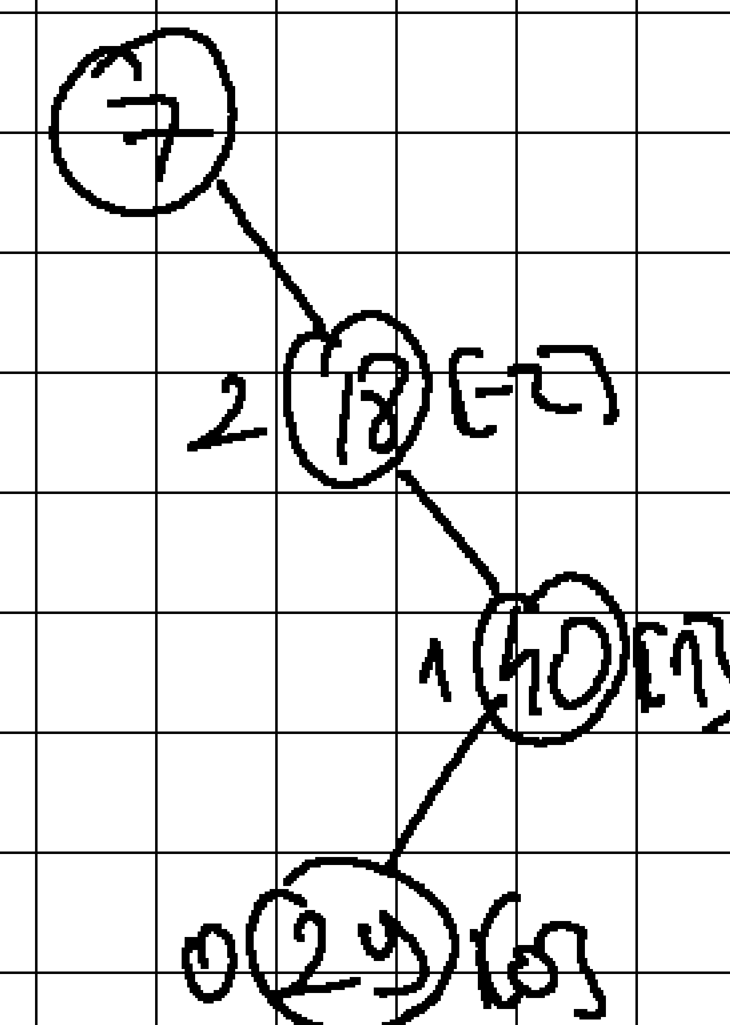
b) $m = 11$



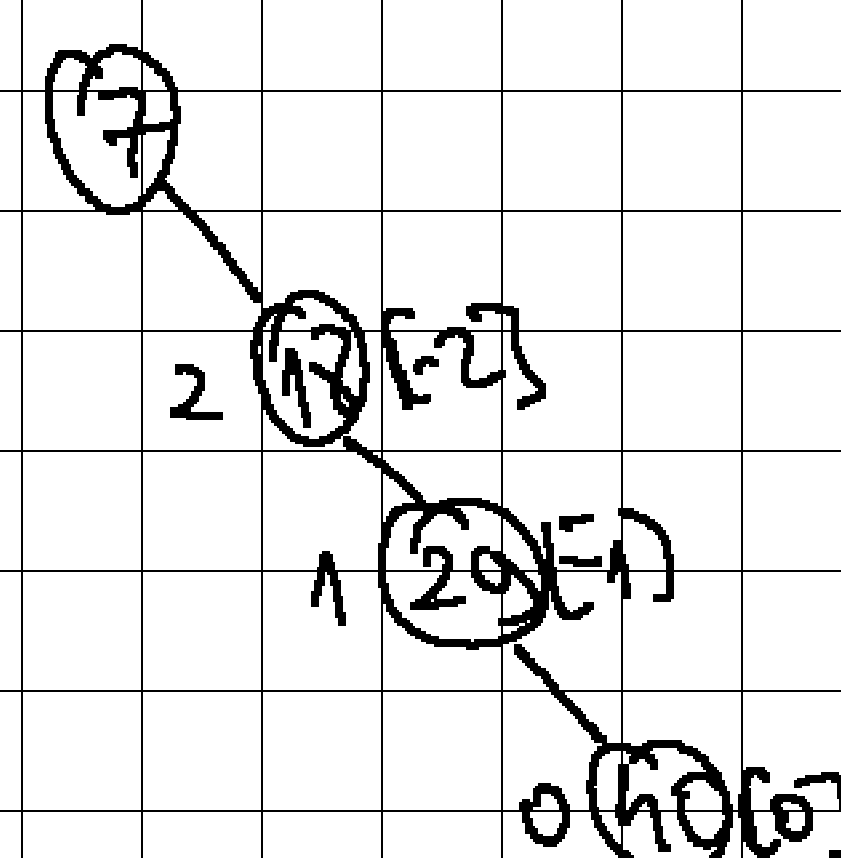
I

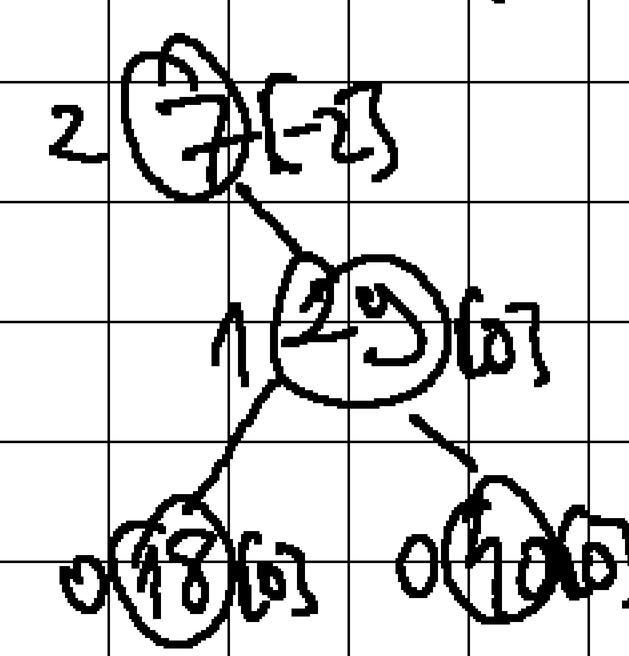


II

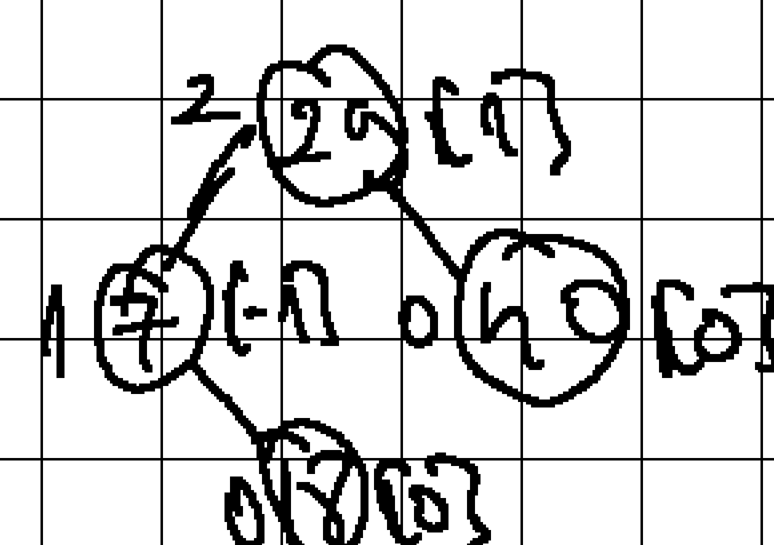


imbalance at node 18 $\Rightarrow$ double left rotation





imbalance at 7 $\Rightarrow$ left rotation



$h(7) = 7 \% 11 = 7$

$h(10) = 10 \% 11 = 10$

$h(120) = 120 \% 11 = 10$

$h(5) = 5 \% 11 = 5$

$h(16) = 16 \% 11 = 5$

$h(18) = 18 \% 11 = 7$

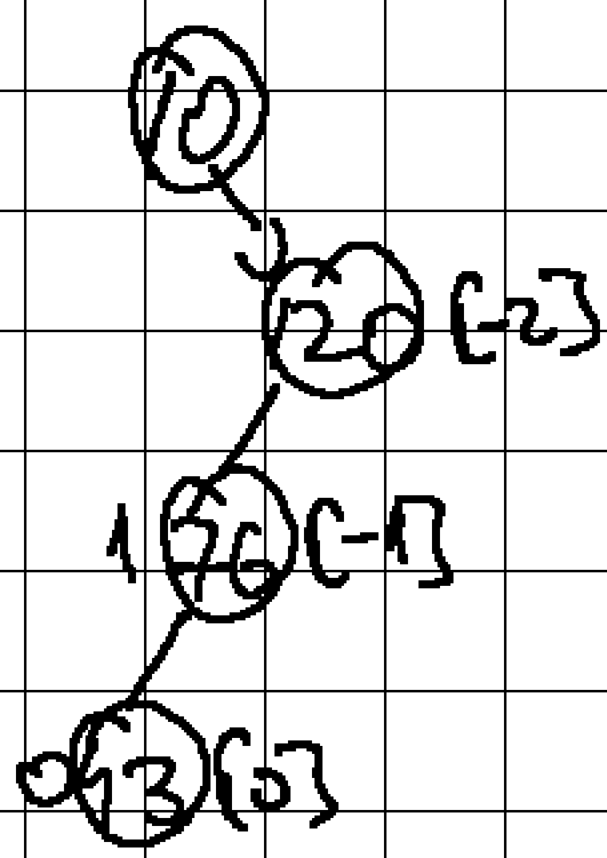
$h(76) = 76 \% 11 = 10$

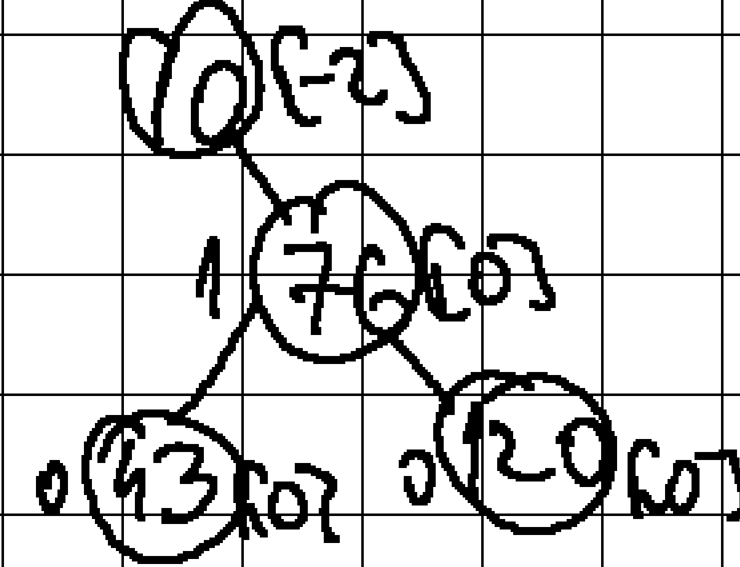
$h(40) = 40 \% 11 = 7$

$h(29) = 29 \% 11 = 7$

$h(43) = 43 \% 11 = 10$

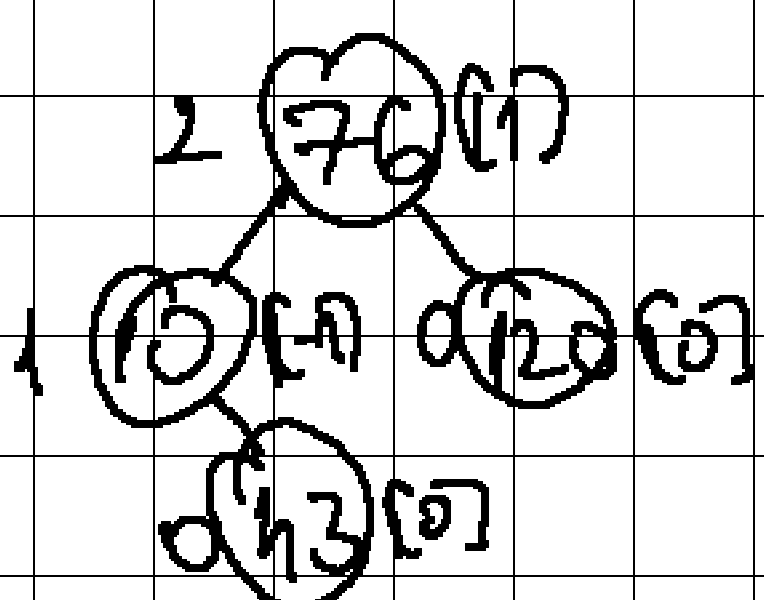
III.





we need a left rotation

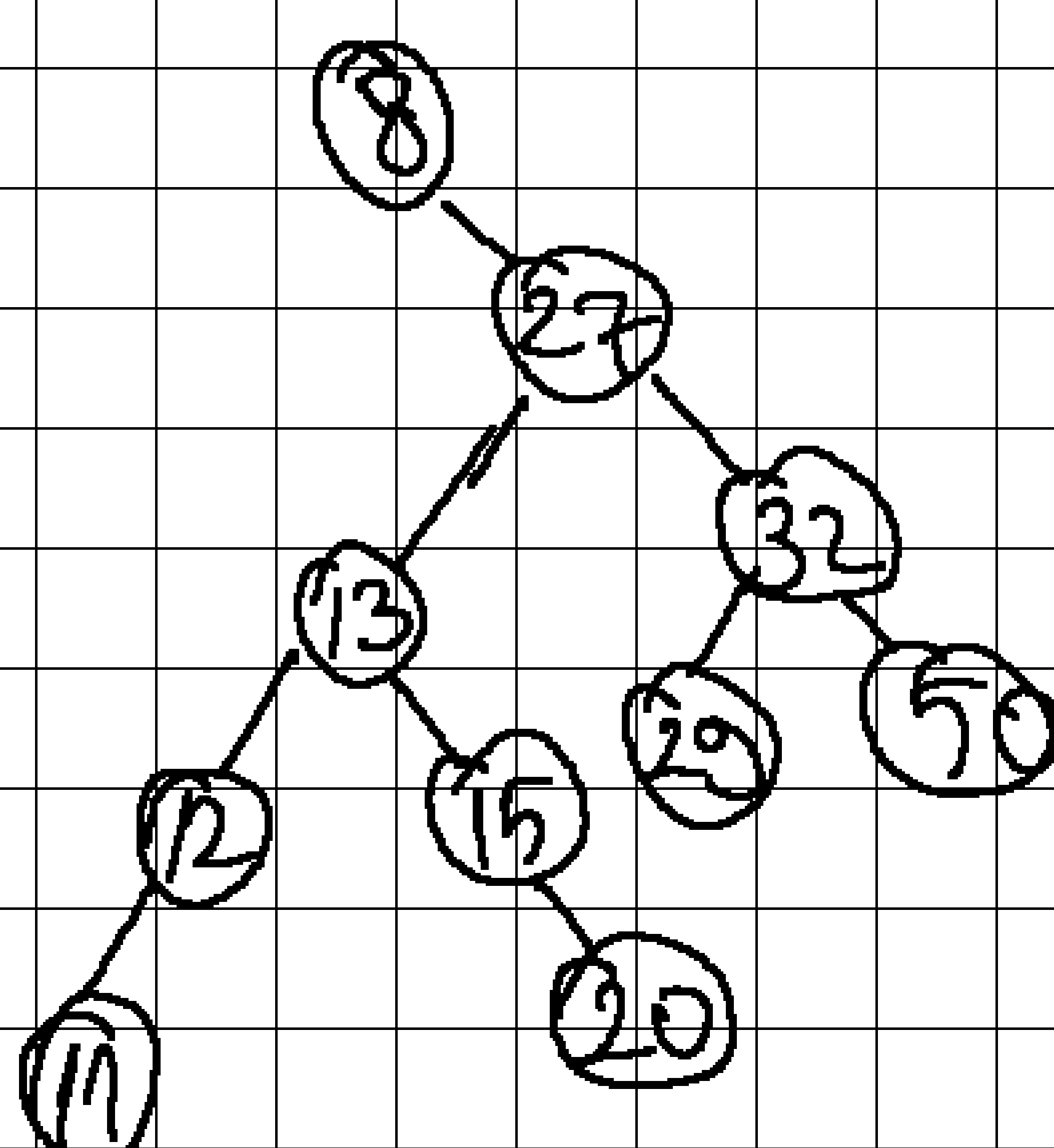
we need a right rotation



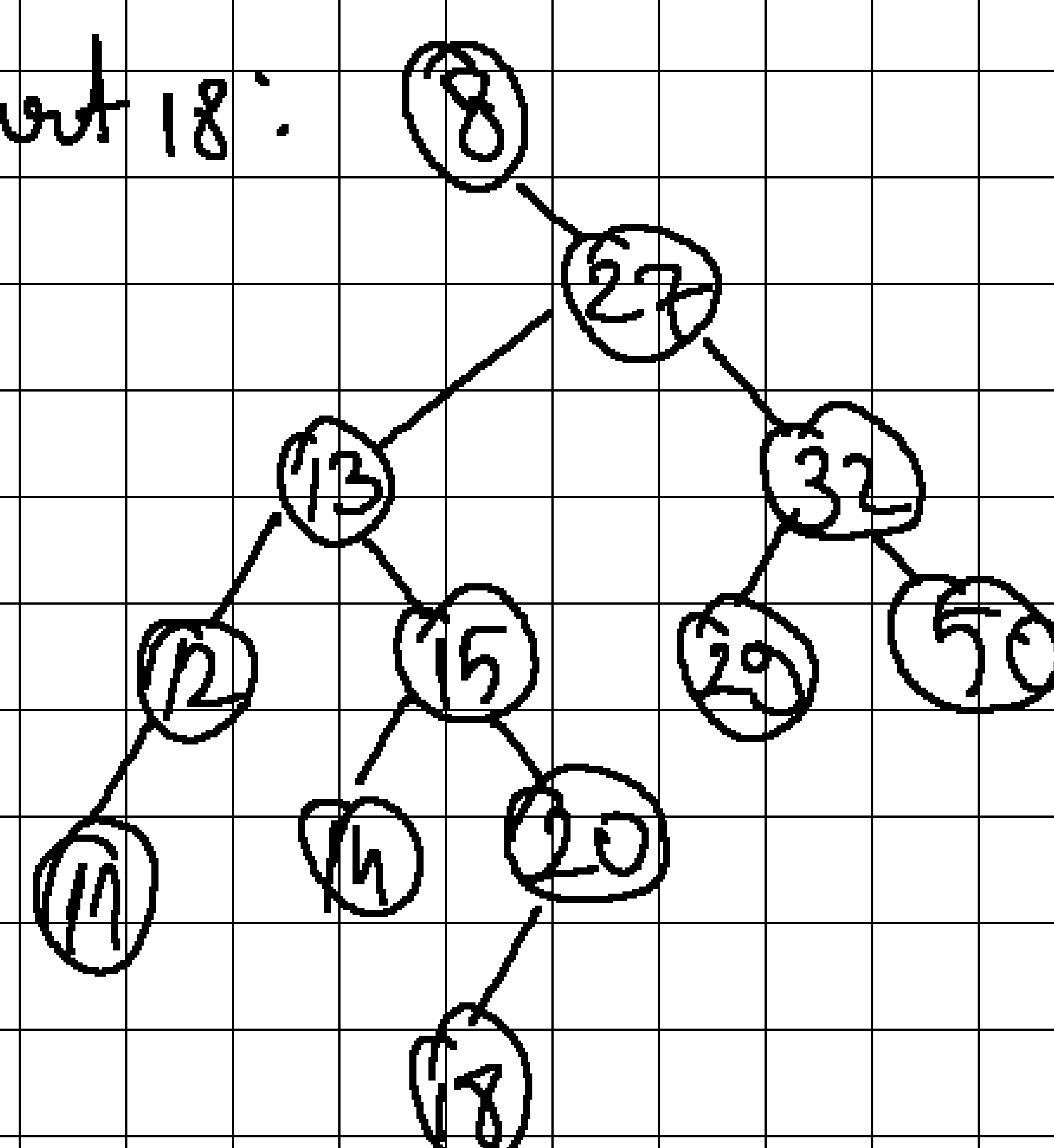
$$\text{load factor} = \frac{\text{no. elems}}{m} = \frac{10}{11}$$

$$\simeq 0{,}9$$

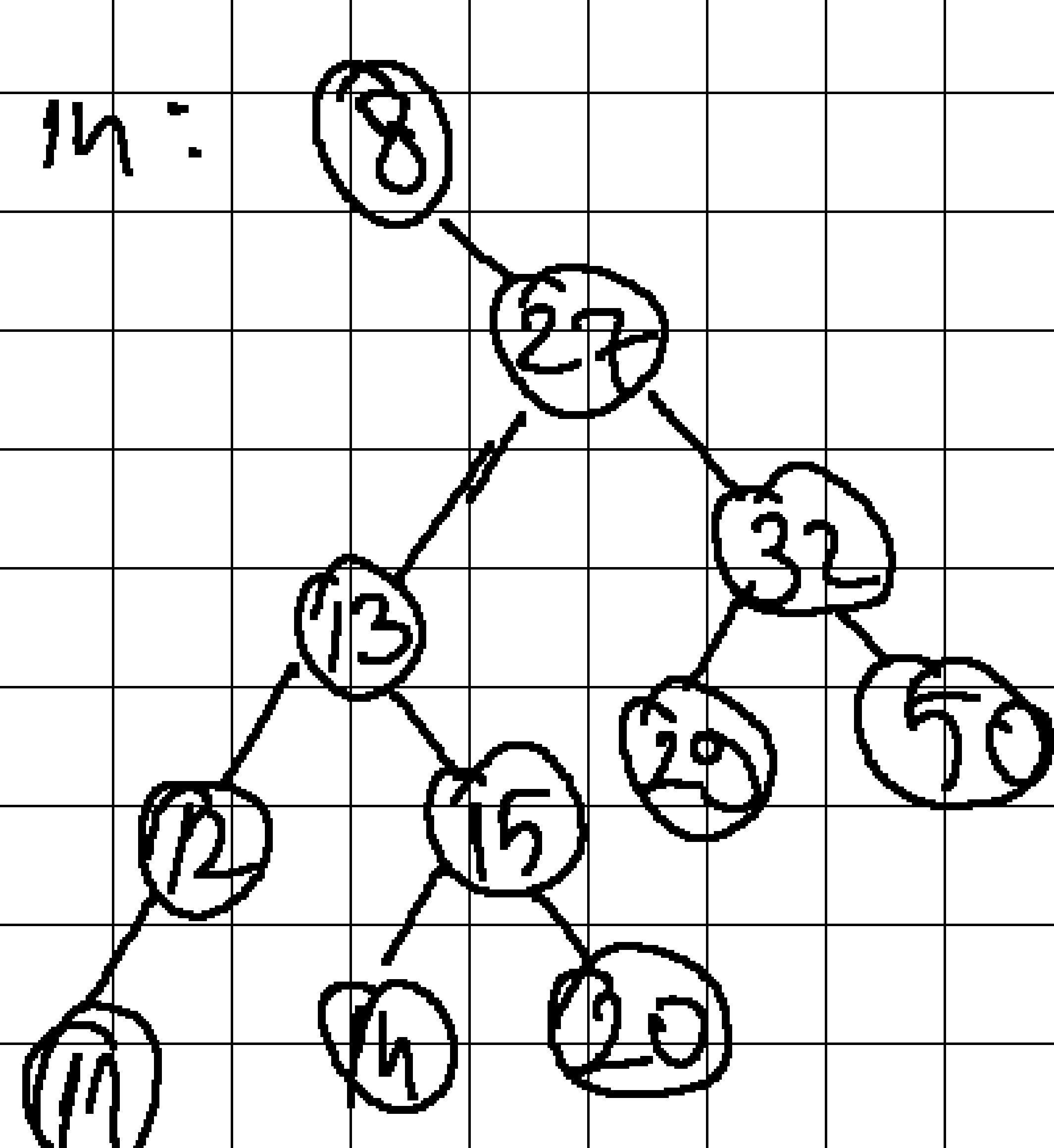
C.



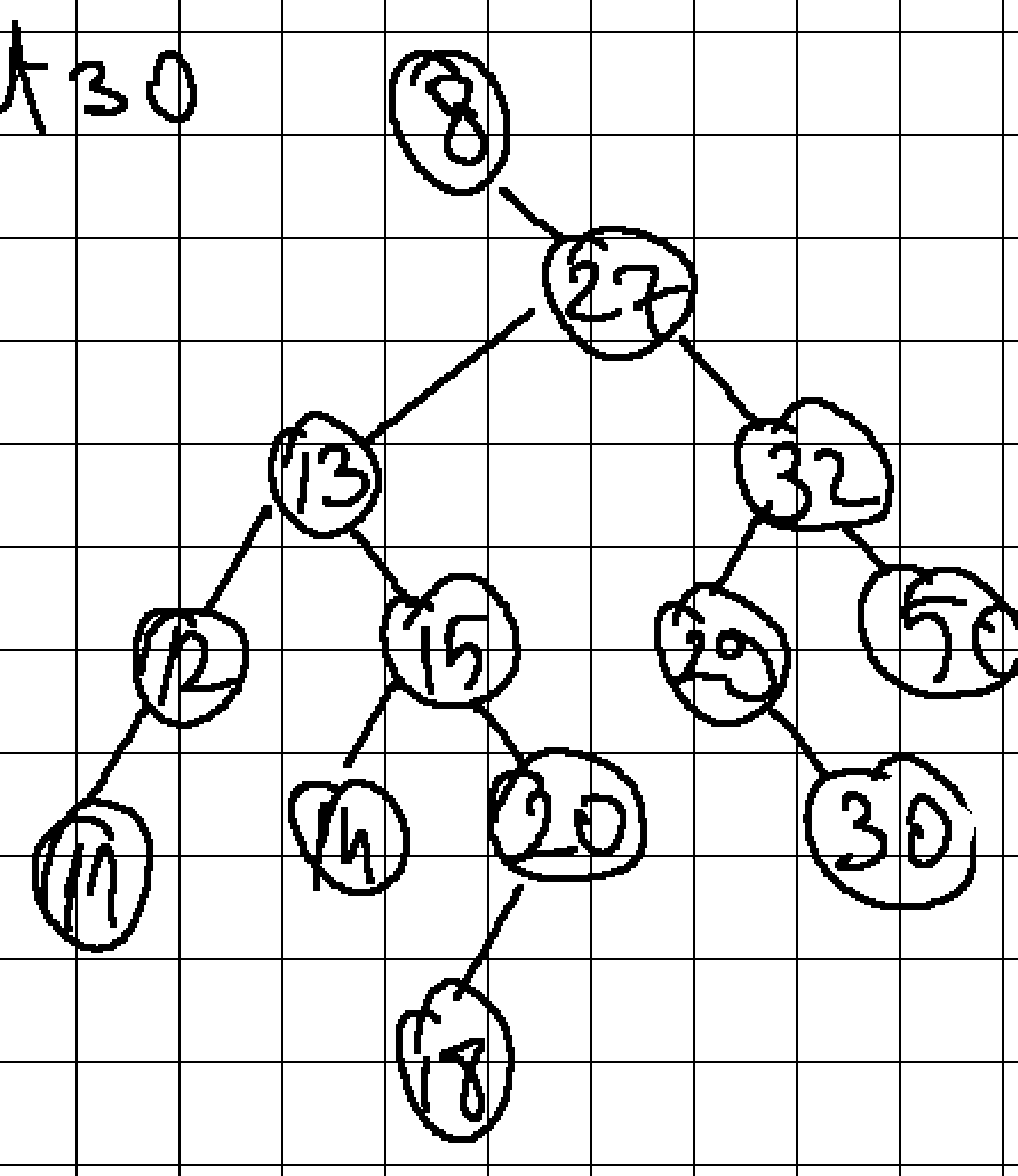
insert 18:



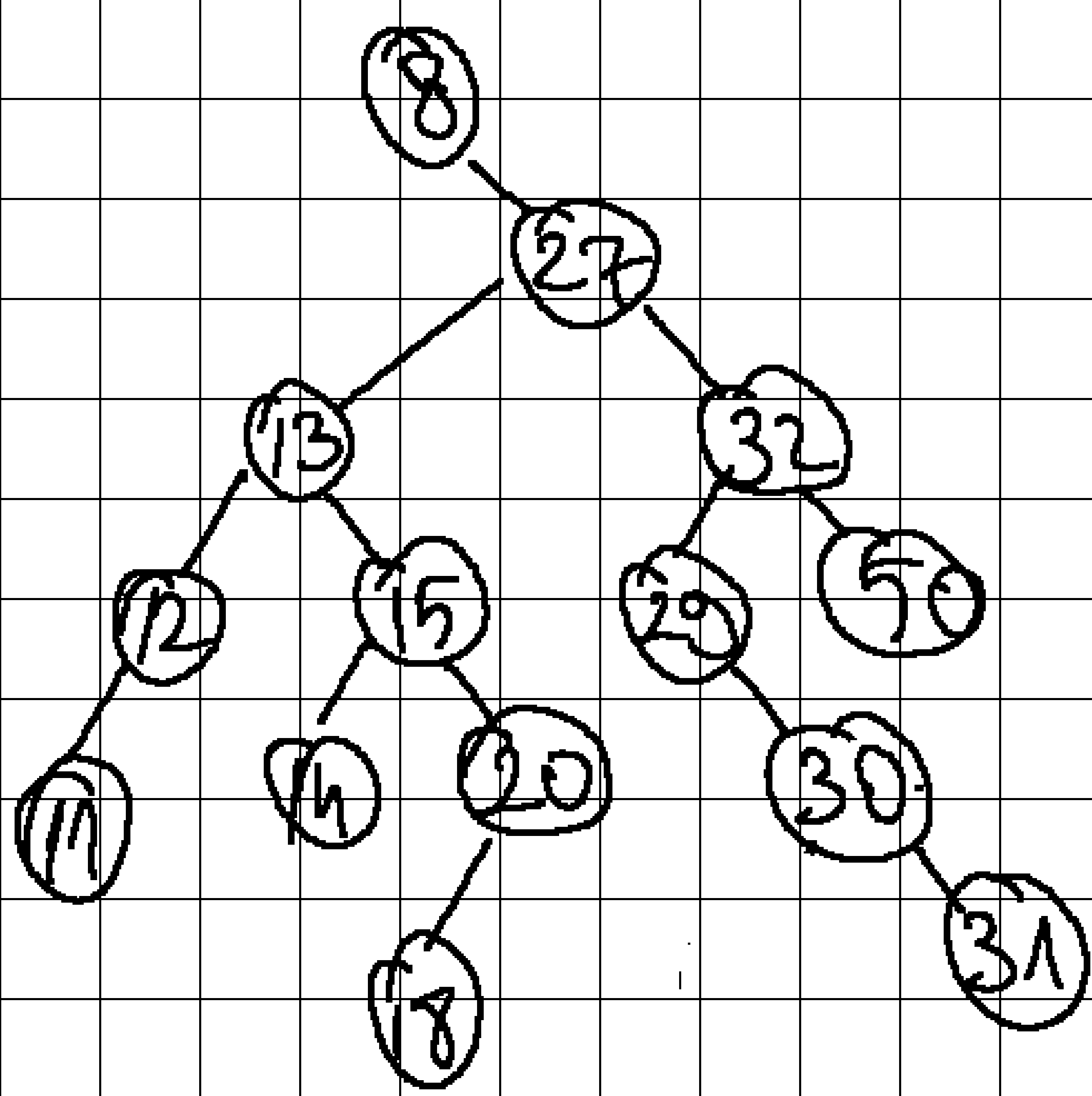
insert 14:



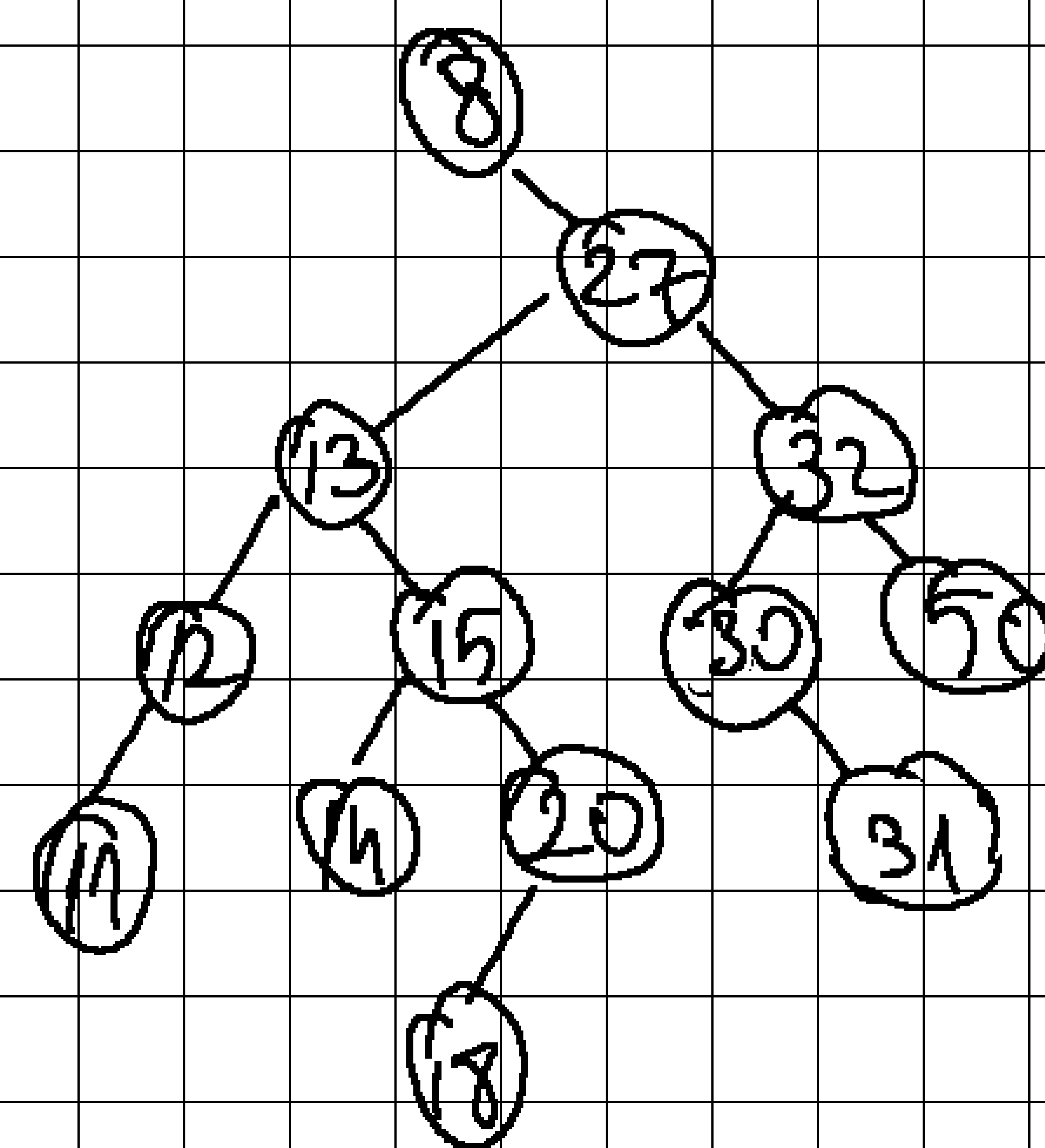
insert 30



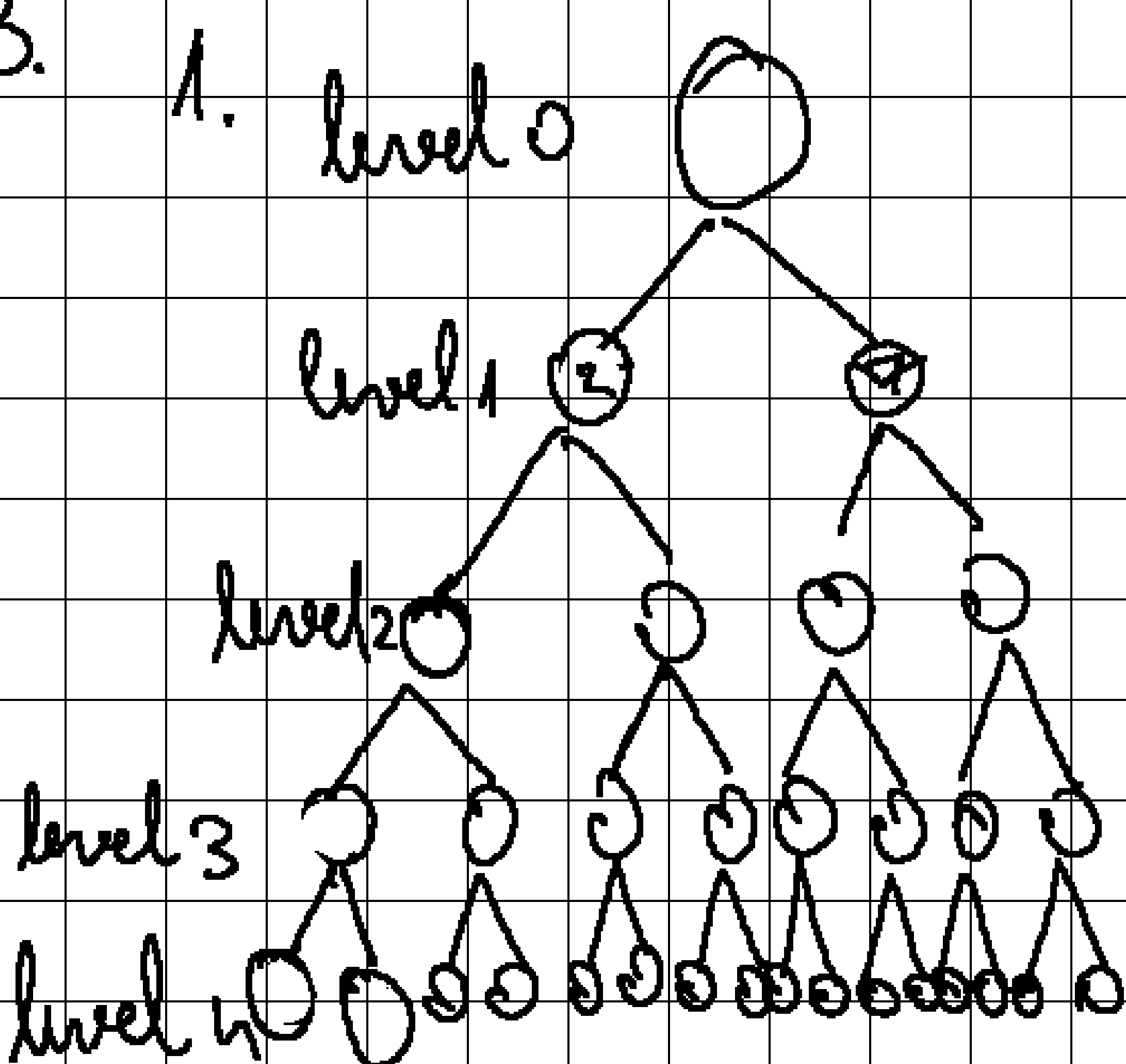
Insert 31



Remove 29:



3. 1.



b. False

it could be on level 1 as well

2. c) we need to know all the elems that will be inserted from the beginning because we need to chose a hash function as optimal as possible to avoid collisions.

3. a, b – False – both can be implemented using an array and linked list.

$\boxed{c}$ queues use two ends of the structure, stacks use only one

4. $N * (\log_2 N) + N^2 * (\log_2 \log_2 N) = x$

$\boxed{a)}$ $x \in \Omega(N * \log_2 N)$ True

$$\lim_{N \to \infty} \frac{N \log_2 N + N^2 (\log_2 \log_2 N)}{N \log_2 N} =$$

$$= \lim_{N \to \infty} \underbrace{\frac{N \cancel{\log_2} N}{\cancel{N \log_2} N}}_{1} + \lim_{N \to \infty} \underbrace{\frac{N^2 (\log_2 \log_2 N)}{\cancel{N} \log_2 N}}_{0} = 1$$

b) $x \in O(N^2)$ False

$$\lim_{N \to \infty} \frac{N \log_2 N + N^2 \log_2 \log_2 N}{N^2} = 0 + \infty = \infty$$

$\boxed{d)}$ $x \in O(N^3)$ True

$$\lim_{N \to \infty} \frac{x}{N^3} = 0$$

c) $x \in \Theta(N^2 * \log_2 N)$ False

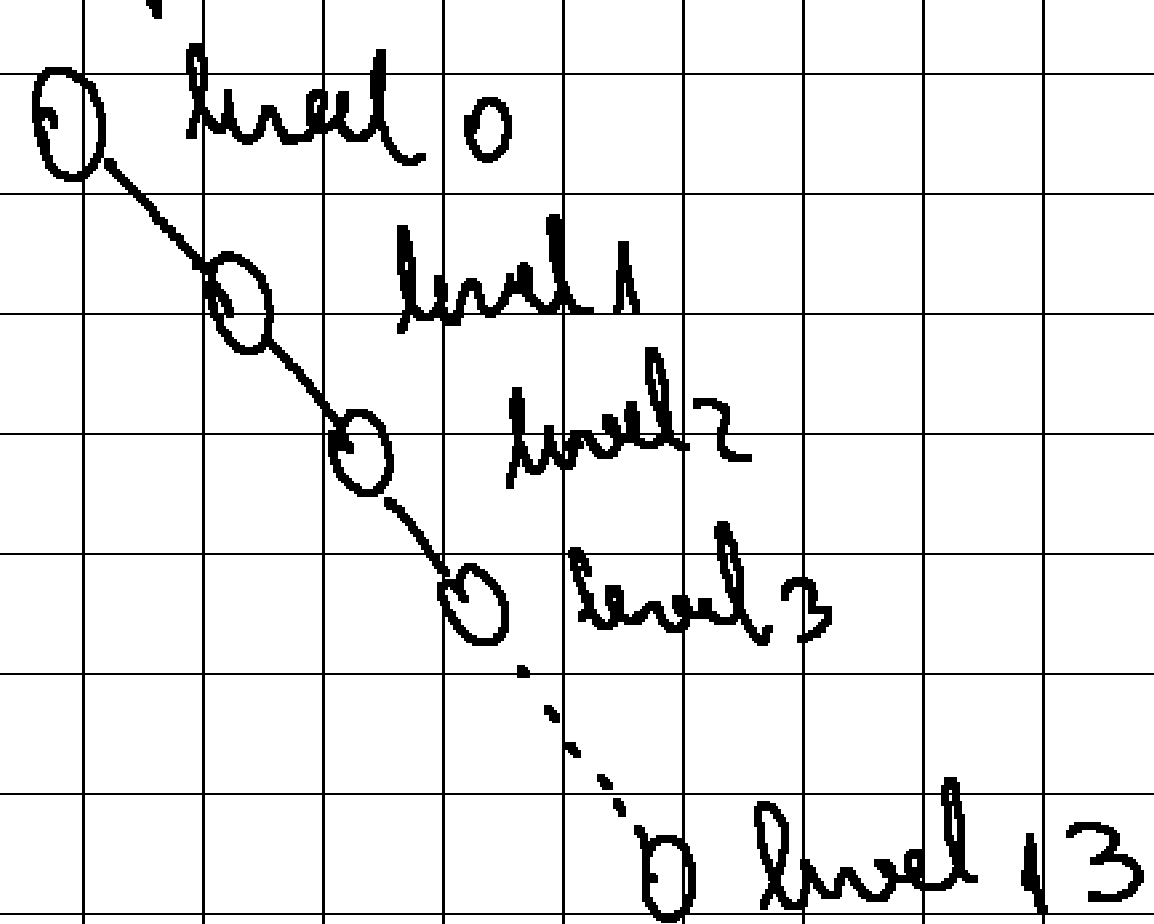
$$\lim_{N \to \infty} \frac{N \log_2 N + N^2 \log_2 \log_2 N}{N^2 \log_2 N} = 0 + 0 = 0$$

5. a.100

Coalesced chaining stores the elems in an array
each elem can store an elem or a link to the
nex elem in the chain

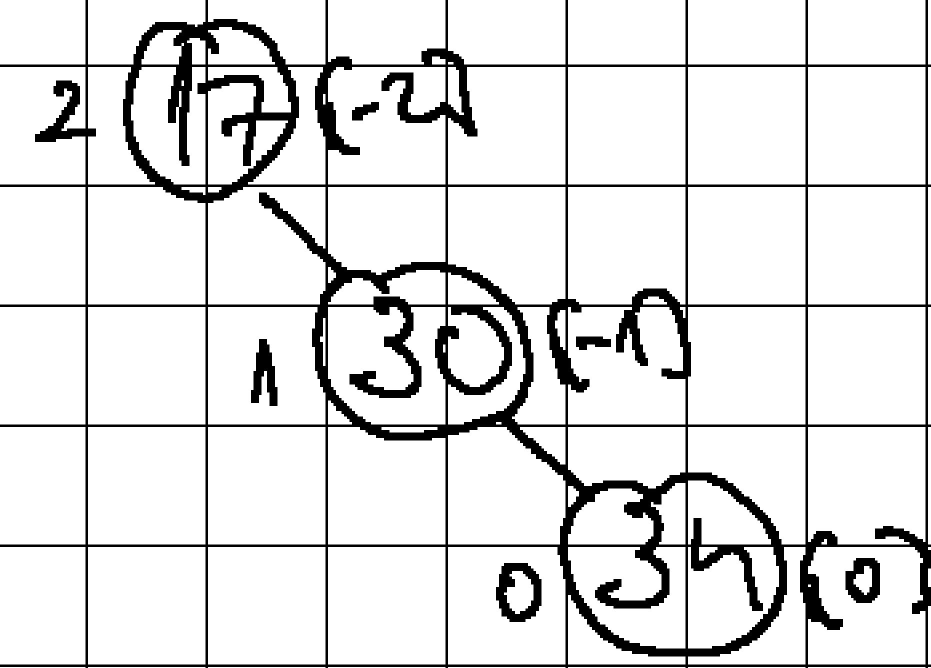
When we hash to the same pos, we go to place
the elem to the next empty pos.

6. depth = 14 − 1 = 13

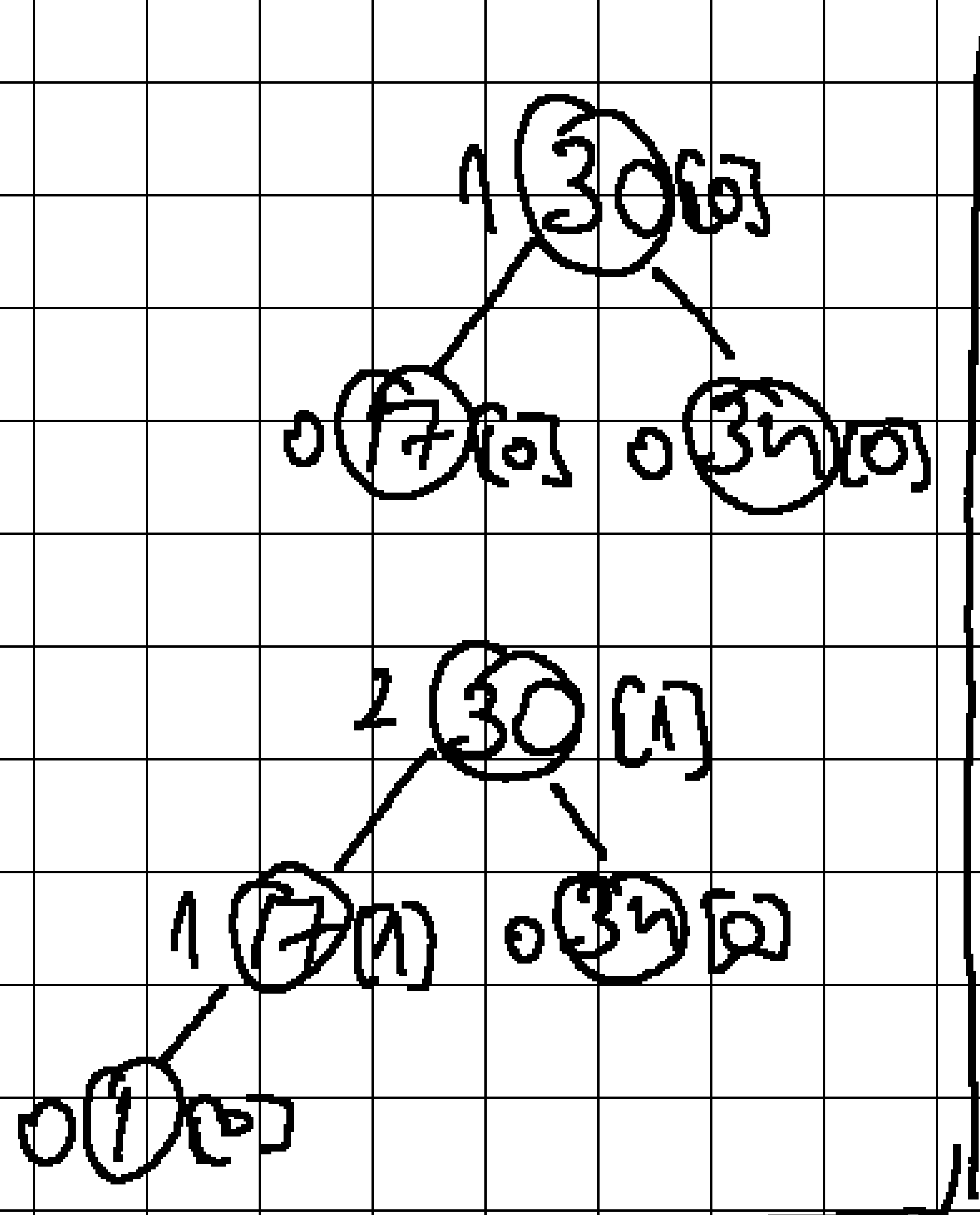


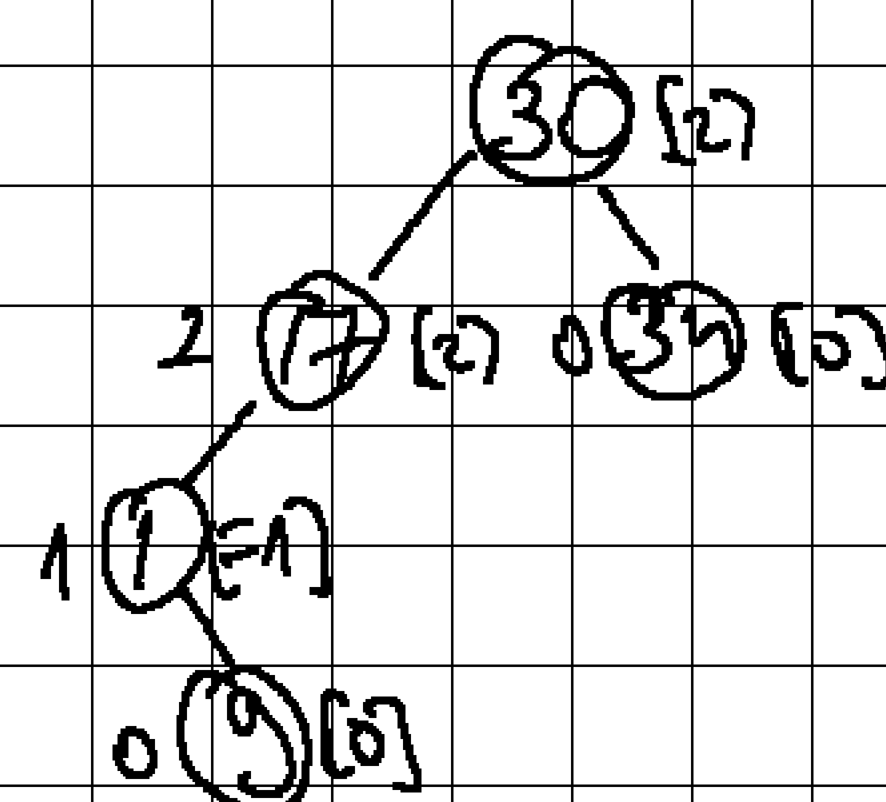
Exam row 17:

2. a)

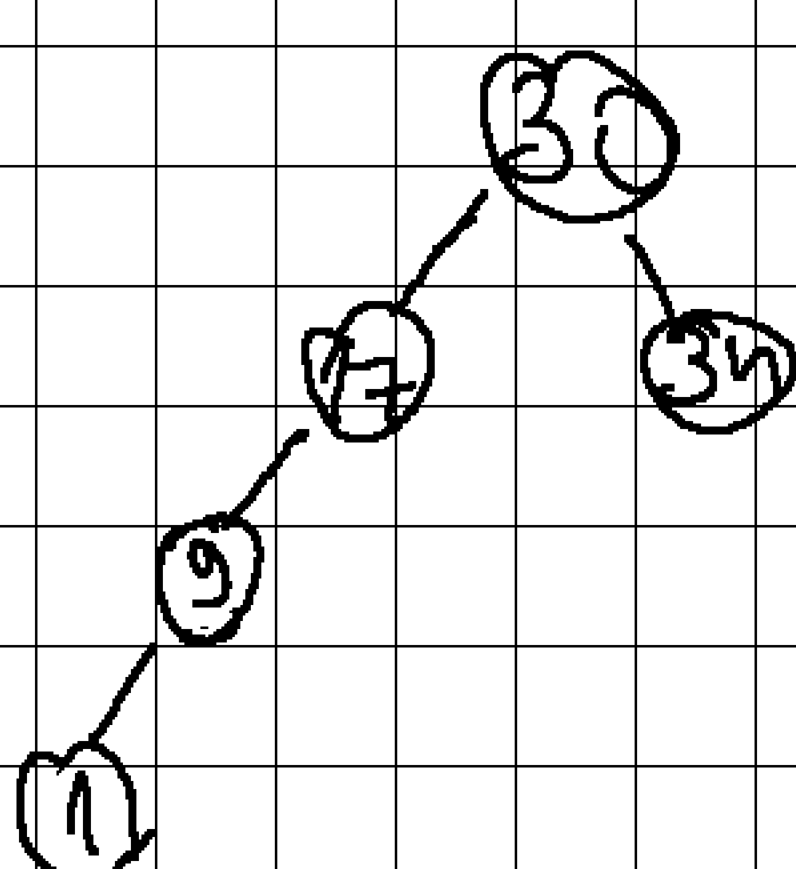


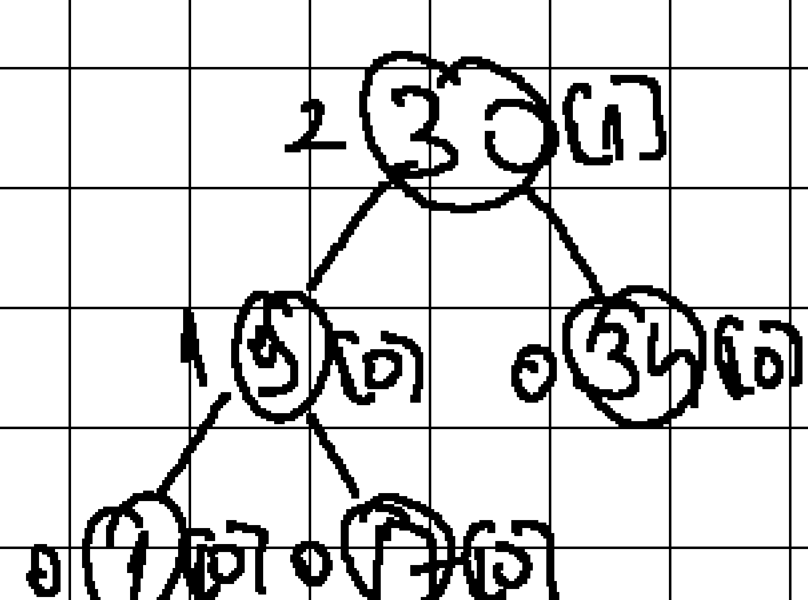
imbalance at node 17, we need a right rotation

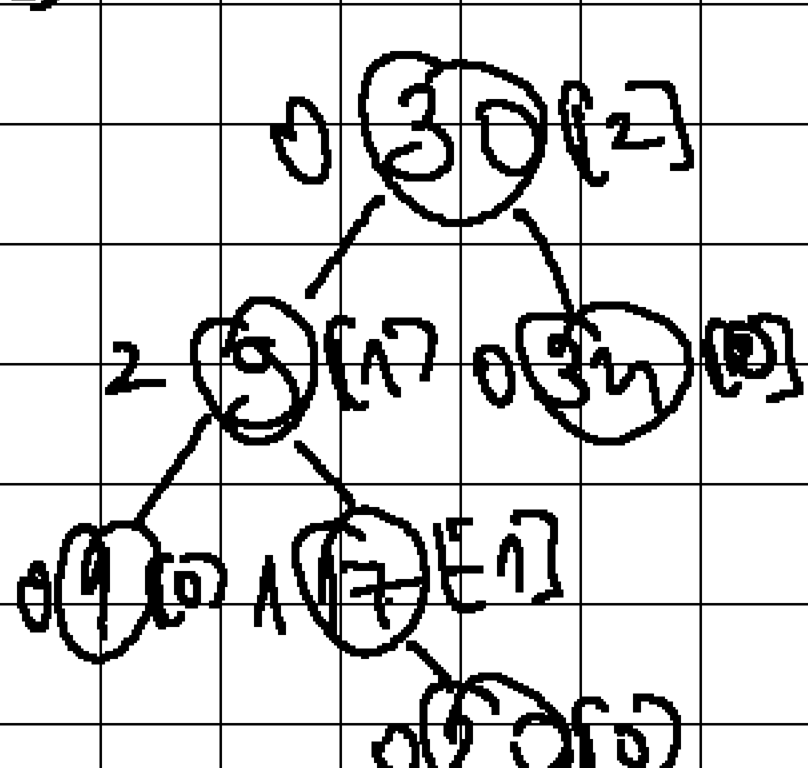




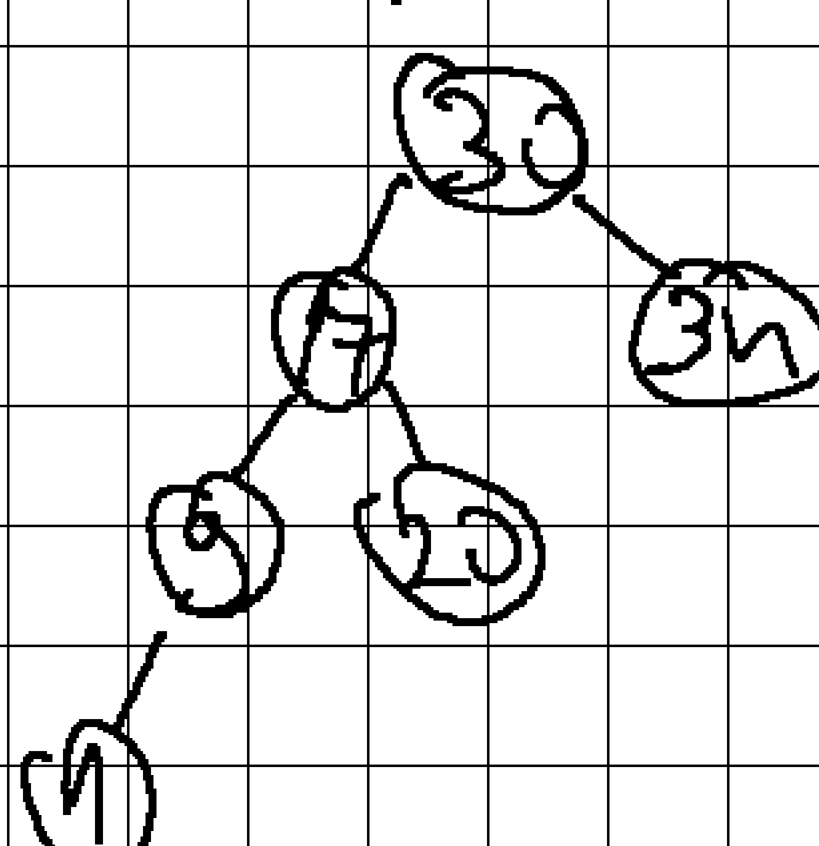
imbalance at node 17 - double right rotation

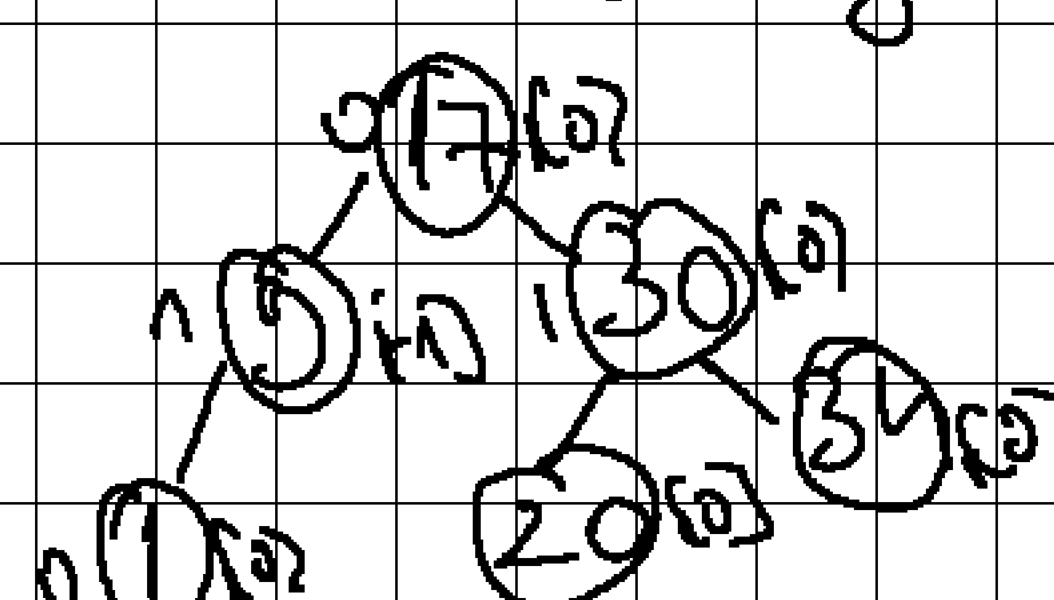


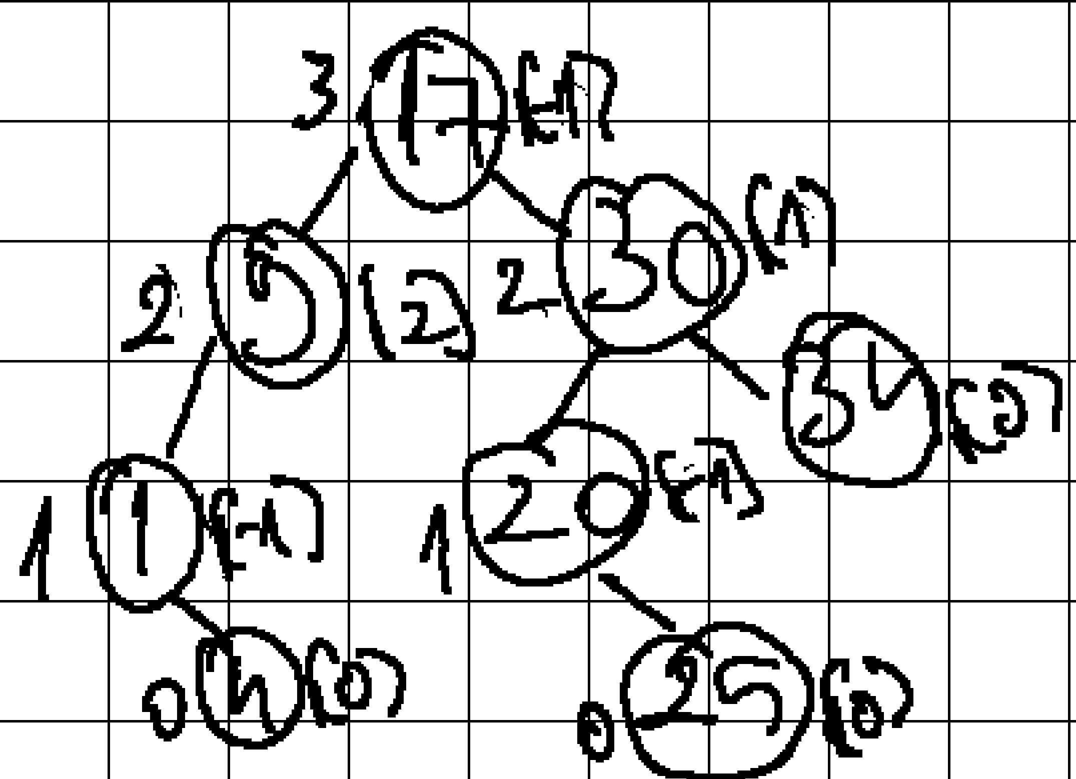




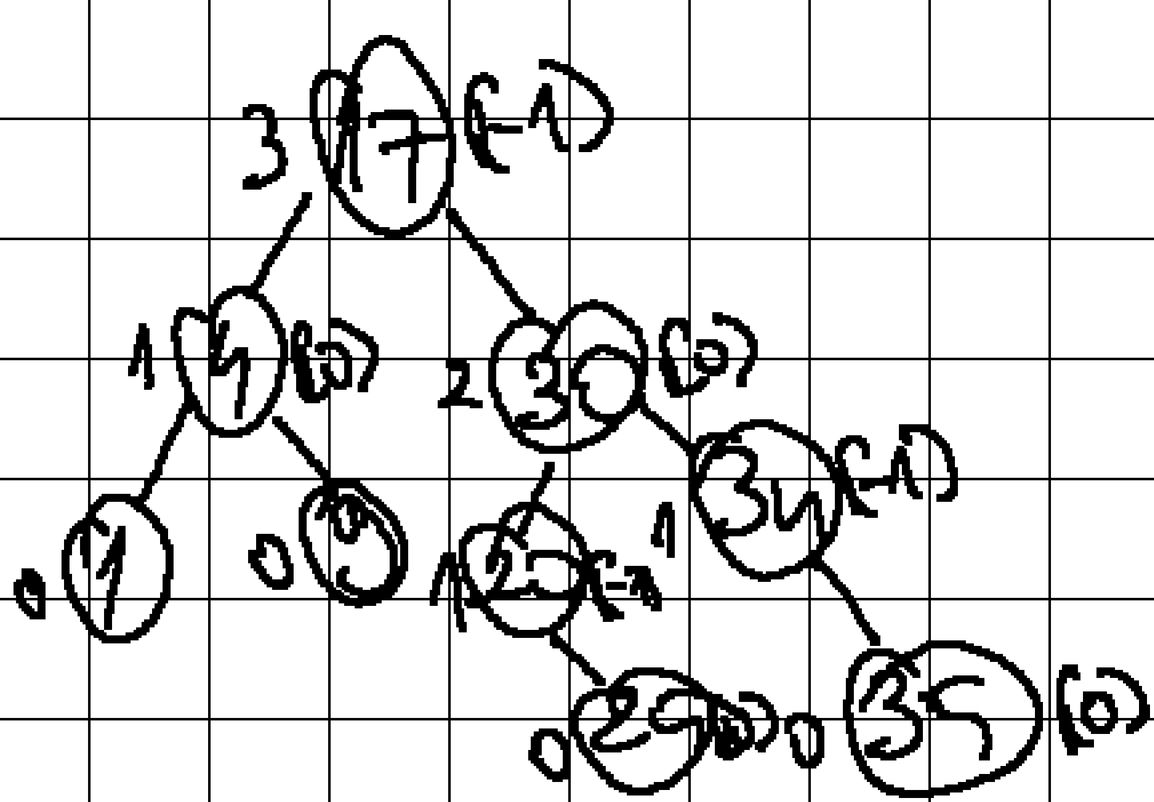
imbalance at 30 ⇒ double right rot

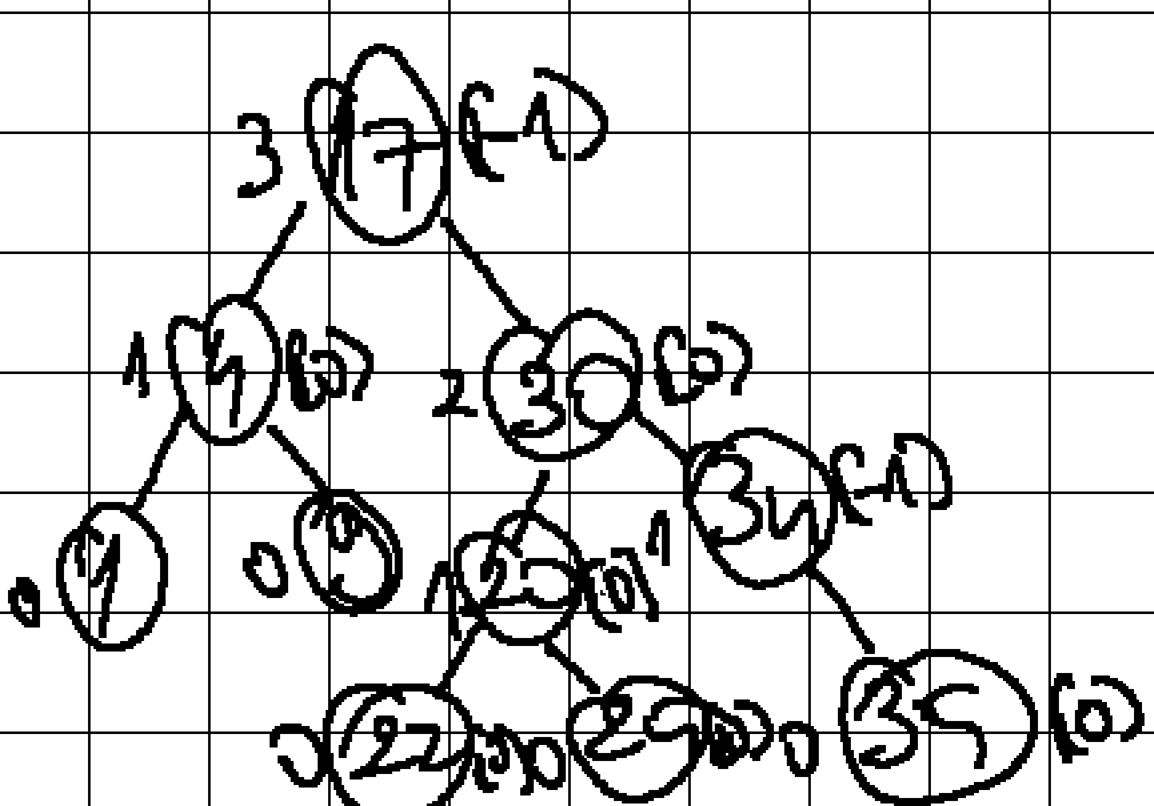




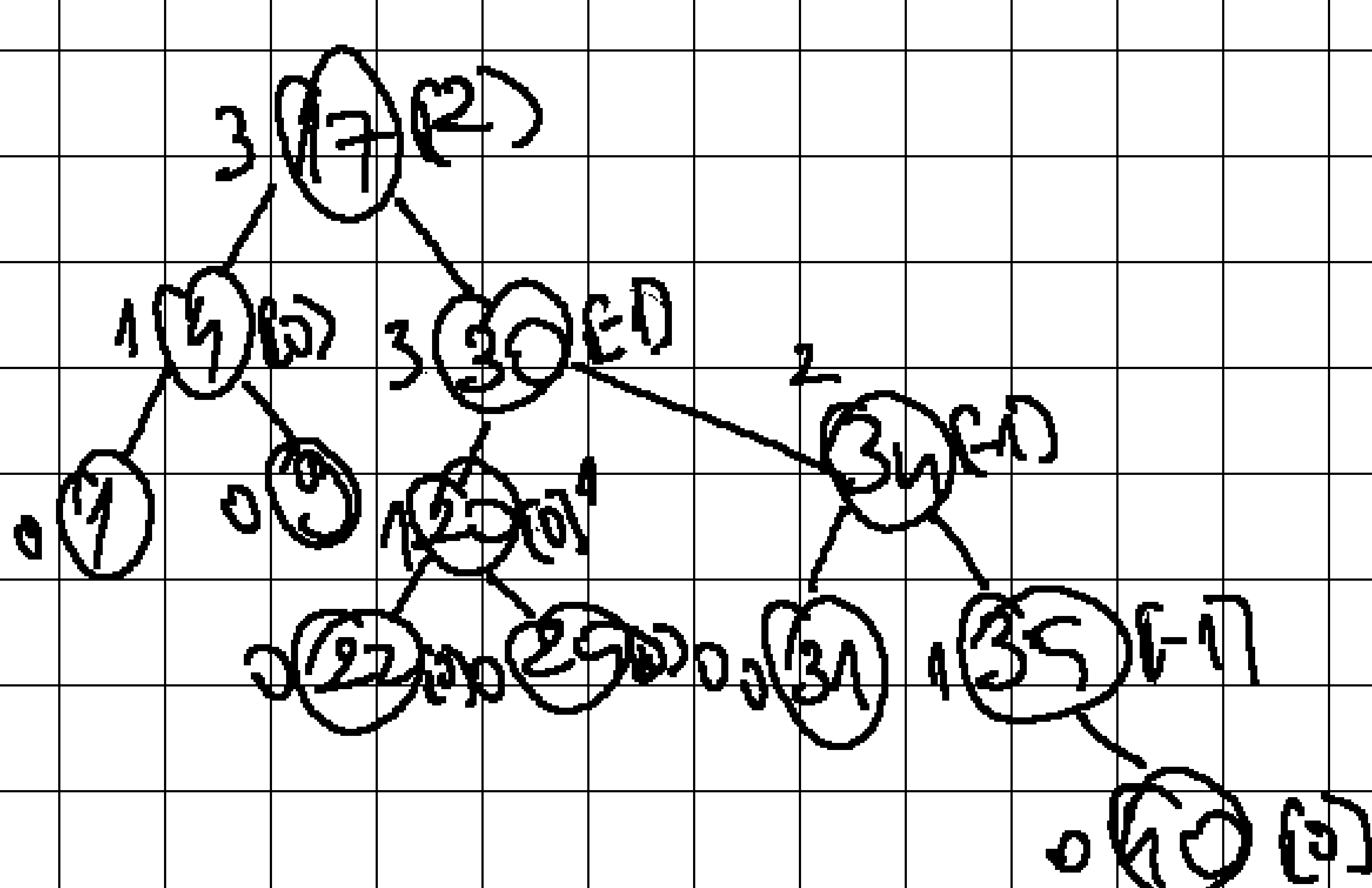


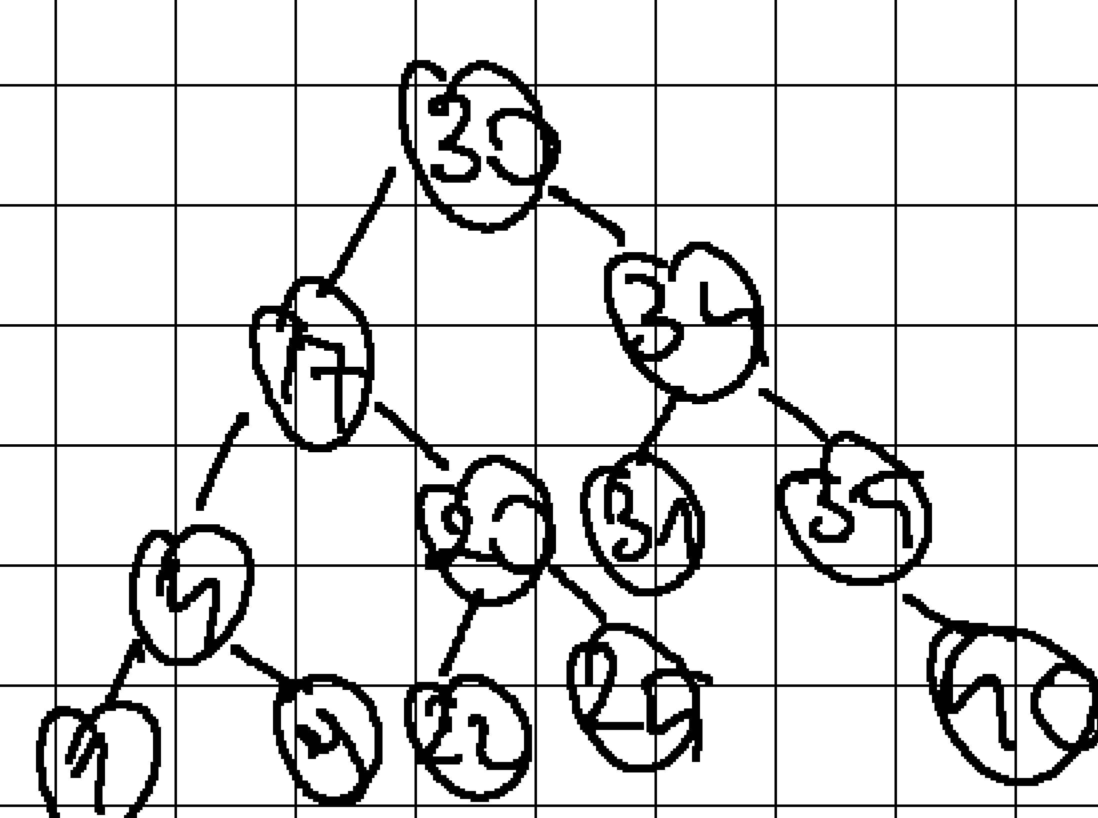
imbalance at node 9 ⇒ double right rotation





imbalance at node 17 ⇒ left rotation





b) $m = 13$

$h(k,i) = (h'(k) + c_1 * i + c_2 * i^2) \% m \qquad h'(k) = k \% m$

$c_1 = 1 \quad c_2 = 2$

0	1	2	3	4	5	6	7	8	9	10	11	12
16	32		42	30		19			136		50	

$h(30,0) = 4 \% 13 = 4$

$h(19,0) = 6 \% 13 = 6$

$h(42,0) = 3 \% 13 = 3$

$h(50,0) = 11 \% 13 = 11$

$h(16,0) = 3 \% 13 = 3$

$h(16,1) = (3 + 1 \cdot 1 + 2 \cdot 1) \% 13 = 6 \% 13 = 6$

$h(16,2) = (3 + 1 \cdot 2 + 2 \cdot 4) \% 13 = 13 \% 13 = 0$

$h(136,0) = 6 \% 13 = 6$

$h(136,1) = (6 + 1 + 2) \% 13 = 9$

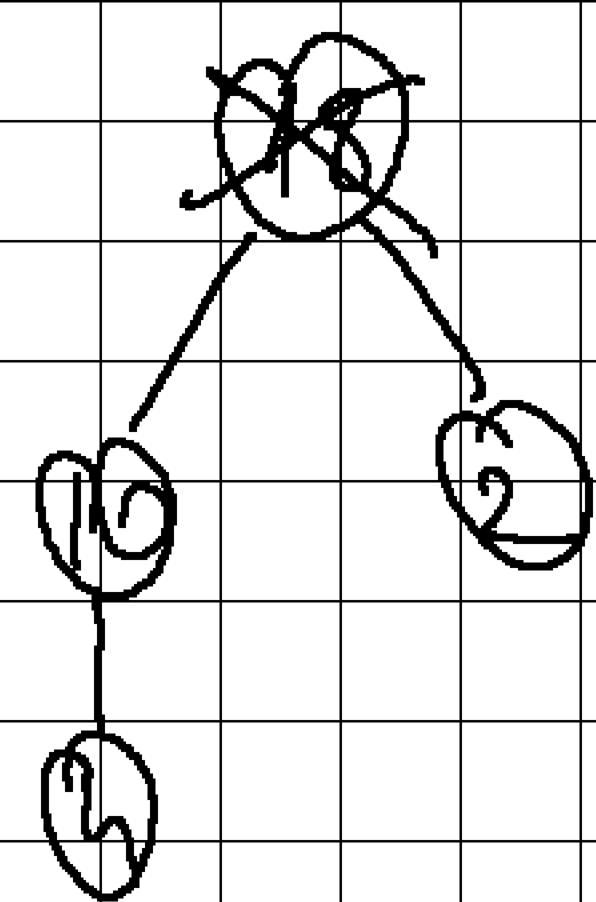
$h(32,0) = 6 \% 13 = 6$

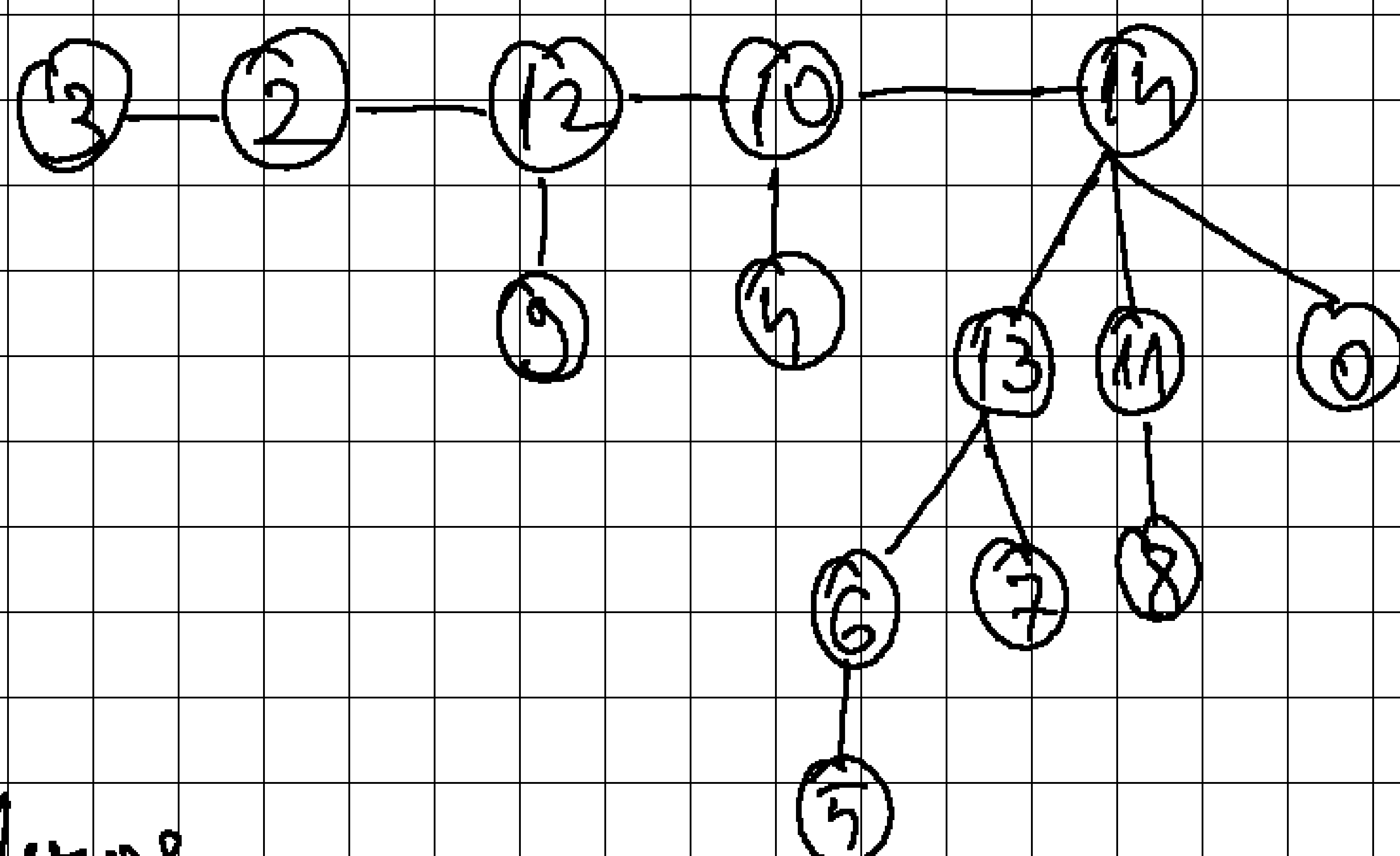
$h(32,1) = (6 + 1 + 2) \% 13 = 9$

$h(32,2) = (6 + 2 + 2 \cdot 4) \% 13 = 16 \% 13 = 3$

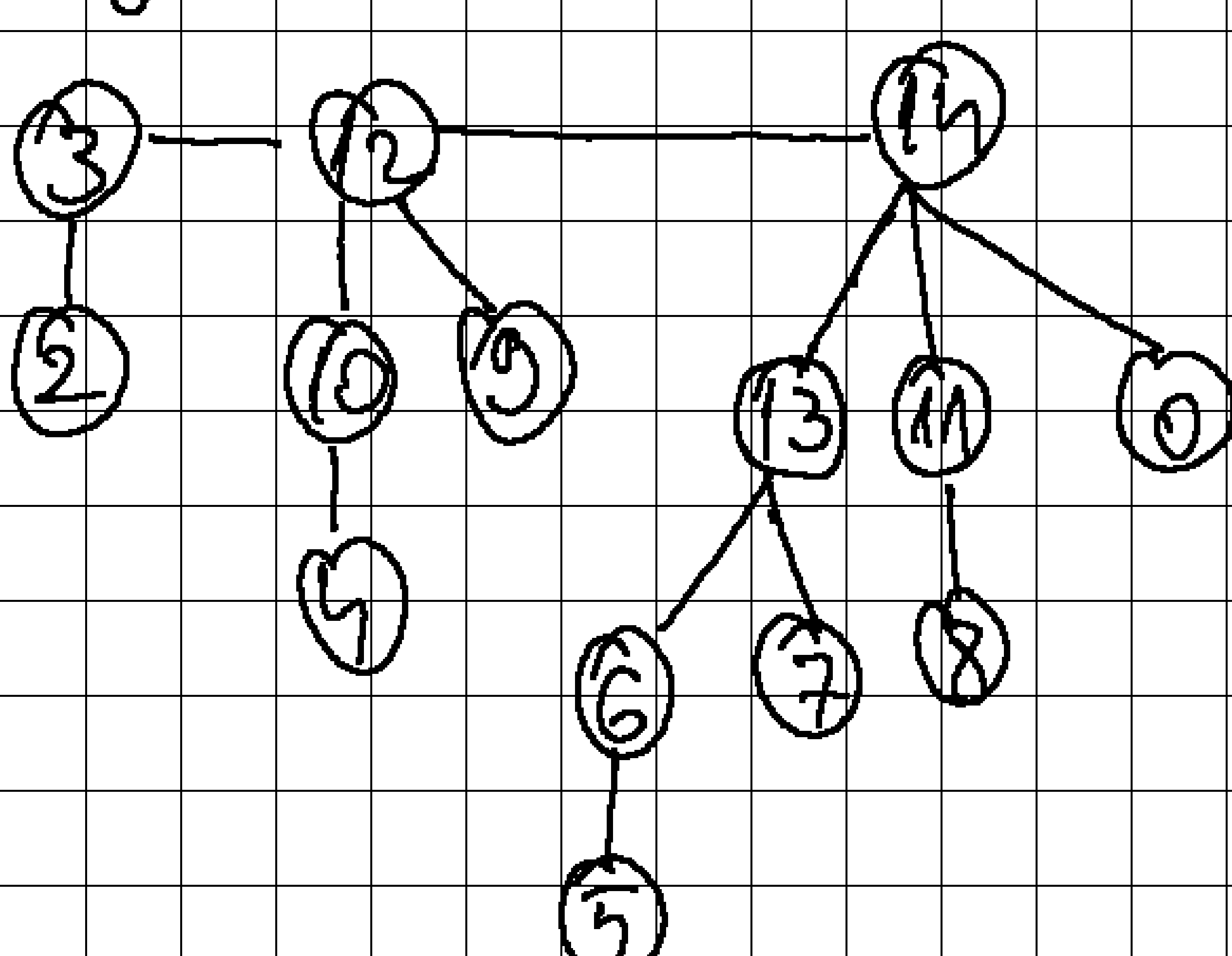
$h(32,3) = (6 + 3 + 18) \% 13 = 27 \% 13 = 1$

c) we see that it's a max heap so we remove the node 18





Merge



3. 1. a. 19

because we do an inorder traversal left root right we are firstly going in the left subtree, and because we have a BST we are going to find 19 first before 30

2. $1 \quad 2 \quad \underbrace{\boxed{3} \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad \boxed{10}}_{8} \quad 11 \quad 12$

b. 8 elements

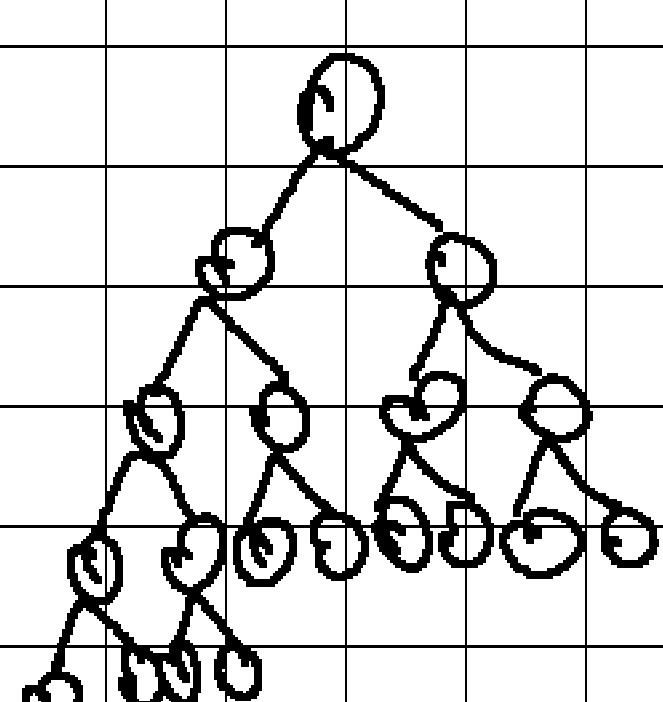
3. the load factor is n/m

where n - number of elems in the table

m - size of the table

low load factor - more empty spaces in the table

4



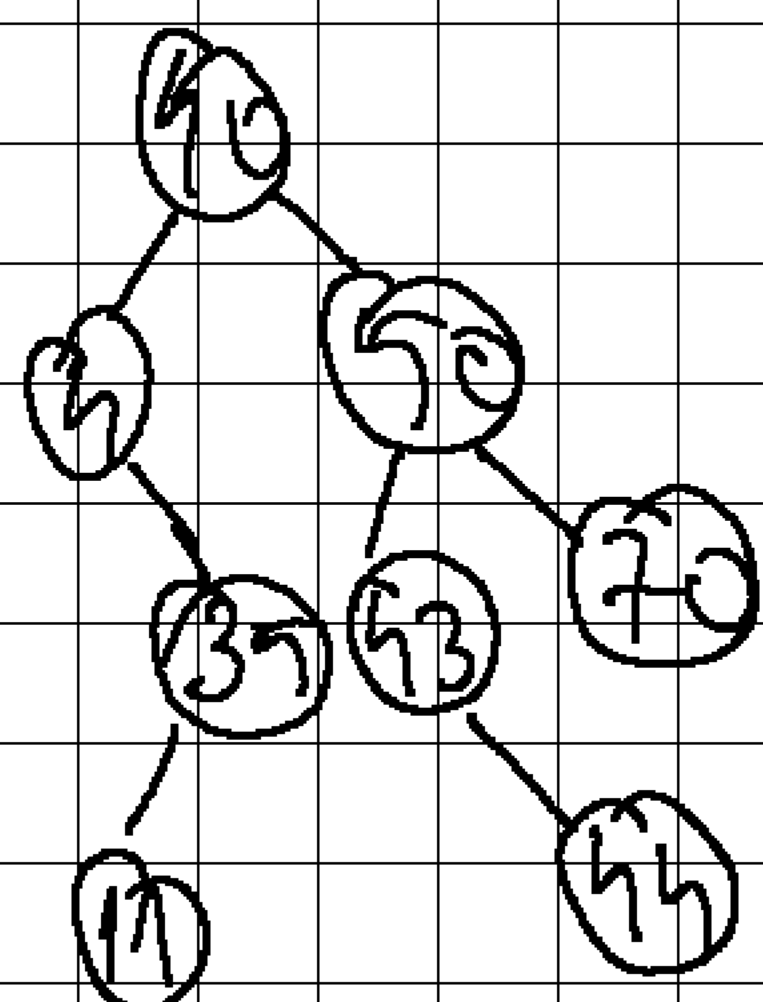
c. 4.

5. b. insert a new key, without a value

– because it does not make sense as maps are designed to store key, value pairs

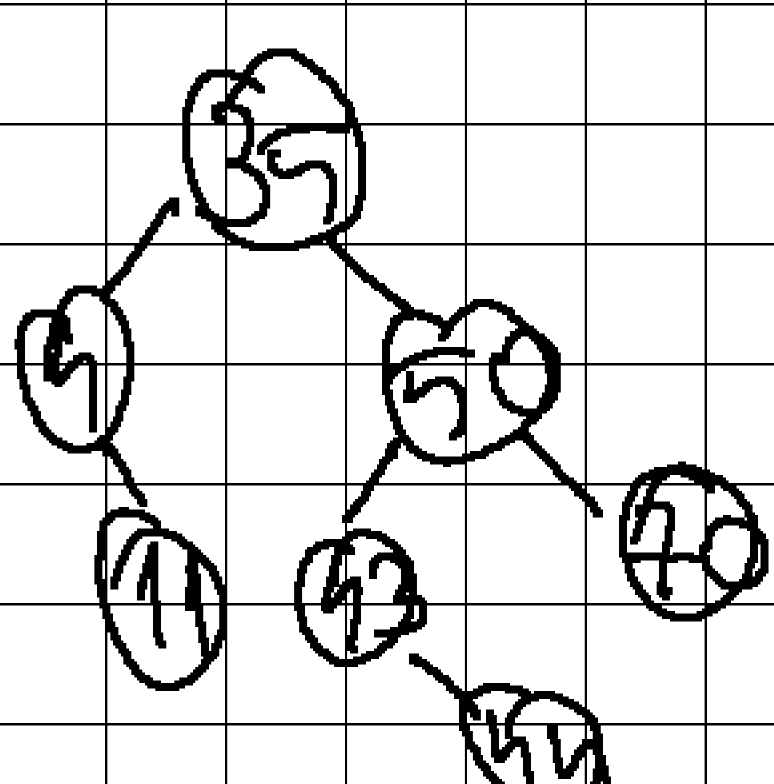
Exam - row 5

2. b)

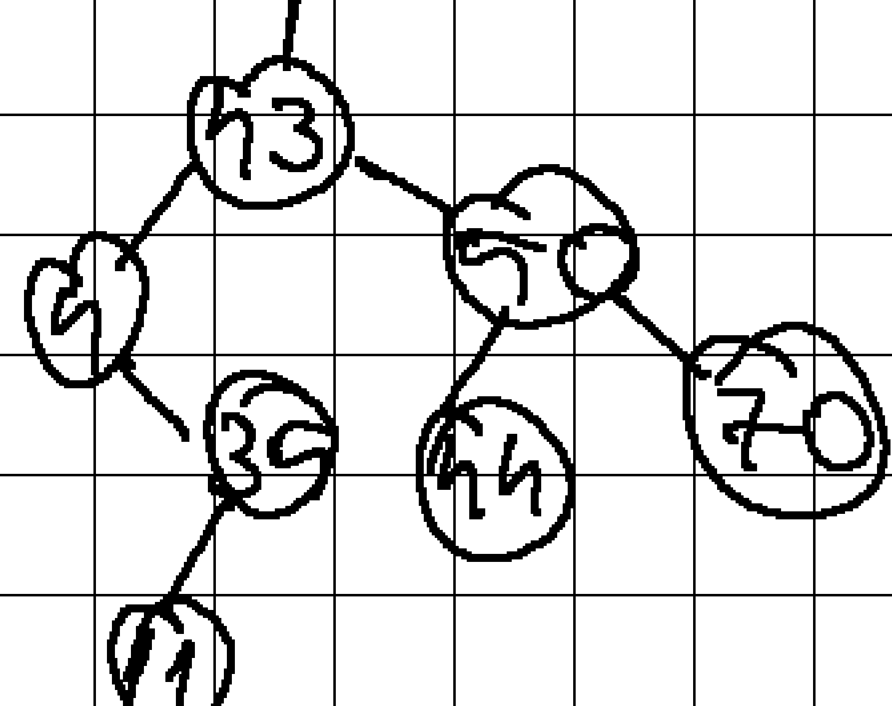


Remove 40:

I we replace it with the max from the left subtree



II We replace it with the min from the right subtree



c) $m = 11$

	0	1	2	3	4	5	6	7	8	9	10
values	84	33	13	29	12	98		7			76
next	1	3	-1	4	-1	-1	-1	0	-1	-1	5

firstEmpty: ~~0~~ ~~1~~ ~~3~~ ~~4~~ 5

$h(7) = 7 \% 11 = 7$

$h(84) = 84 \% 11 = 7$

$h(13) = 13 \% 11 = 2$

$h(33) = 33 \% 11 = 0$

$h(76) = 76 \% 11 = 10$

$h(29) = 29 \% 11 = 7$

$h(12) = 12 \% 11 = 1$

$h(98) = 98 \% 11 = 10$

If we remove value 7, we have to find another value from the links that hashes to that position and replace it with that, and update the next link

	0	1	2	3	4	5	6	7	8	9	10
values		33	13	29	12	98		84			76
next	-1	3	-1	4	-1	-1	-1	1	-1	-1	5

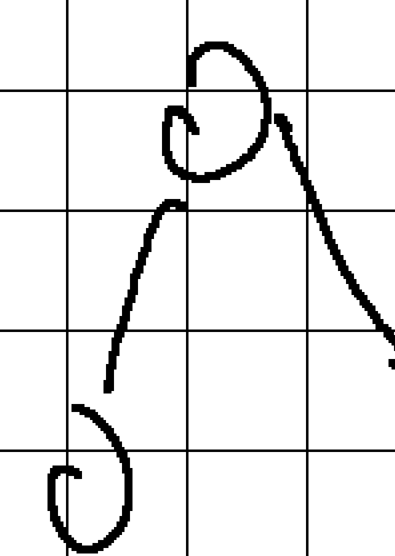
3. 1. a. preorder & inorder

c. Inorder & postorder

preorder & postorder can generate the same tree

2. (( ) (( )) (( )))

stack: ((((((



~~2~~ 3 C. 3

3. e. separate chaining

the load factor can be higher then 1 because we can have chains of elements on the same pos (we are using linked lists)

4. b) the 3rd elem. can be either on level 1 or 2

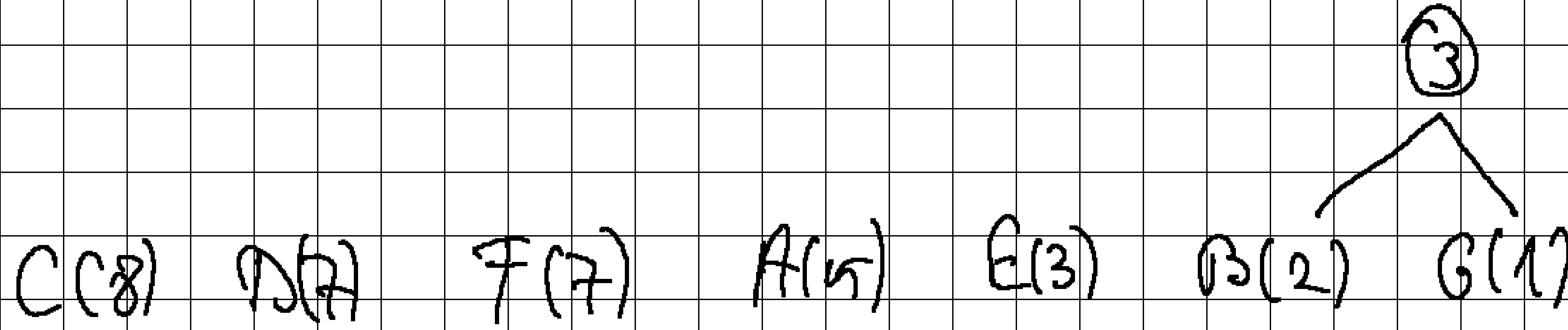
5. d) Modify a currently zero value into zero

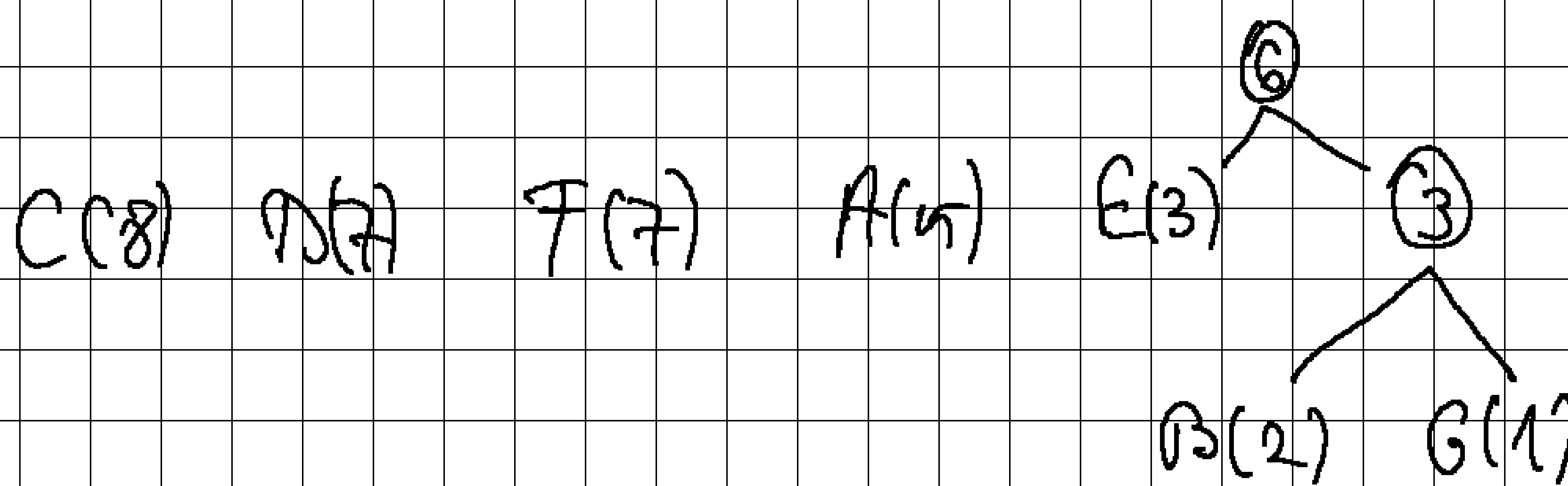
6. $N(h) = N(h-1) + N(h-2) + 1$ C. 12

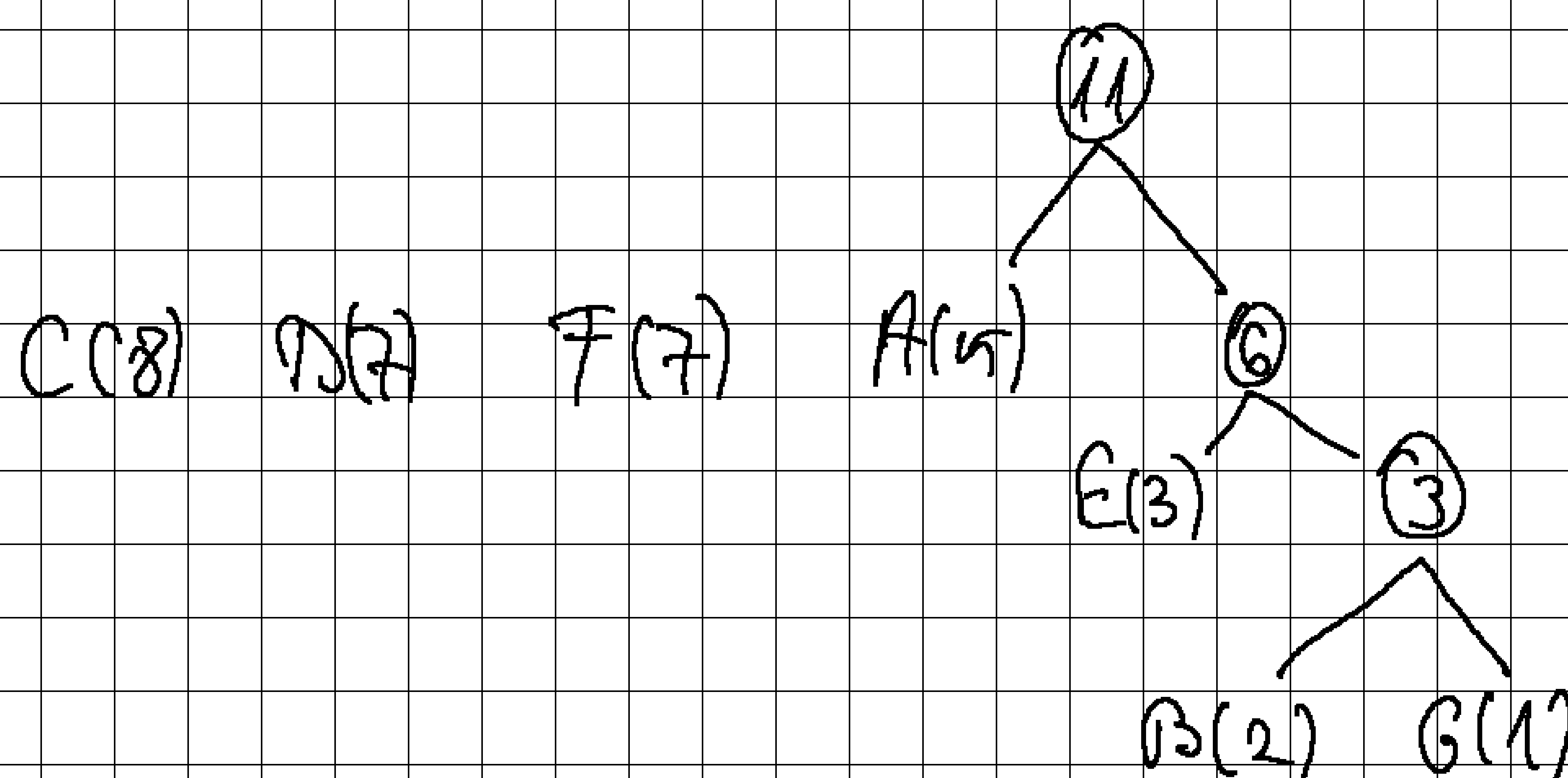
$N(0) = 1$ $N(4) = N(3) + N(2) + 1 = 12$

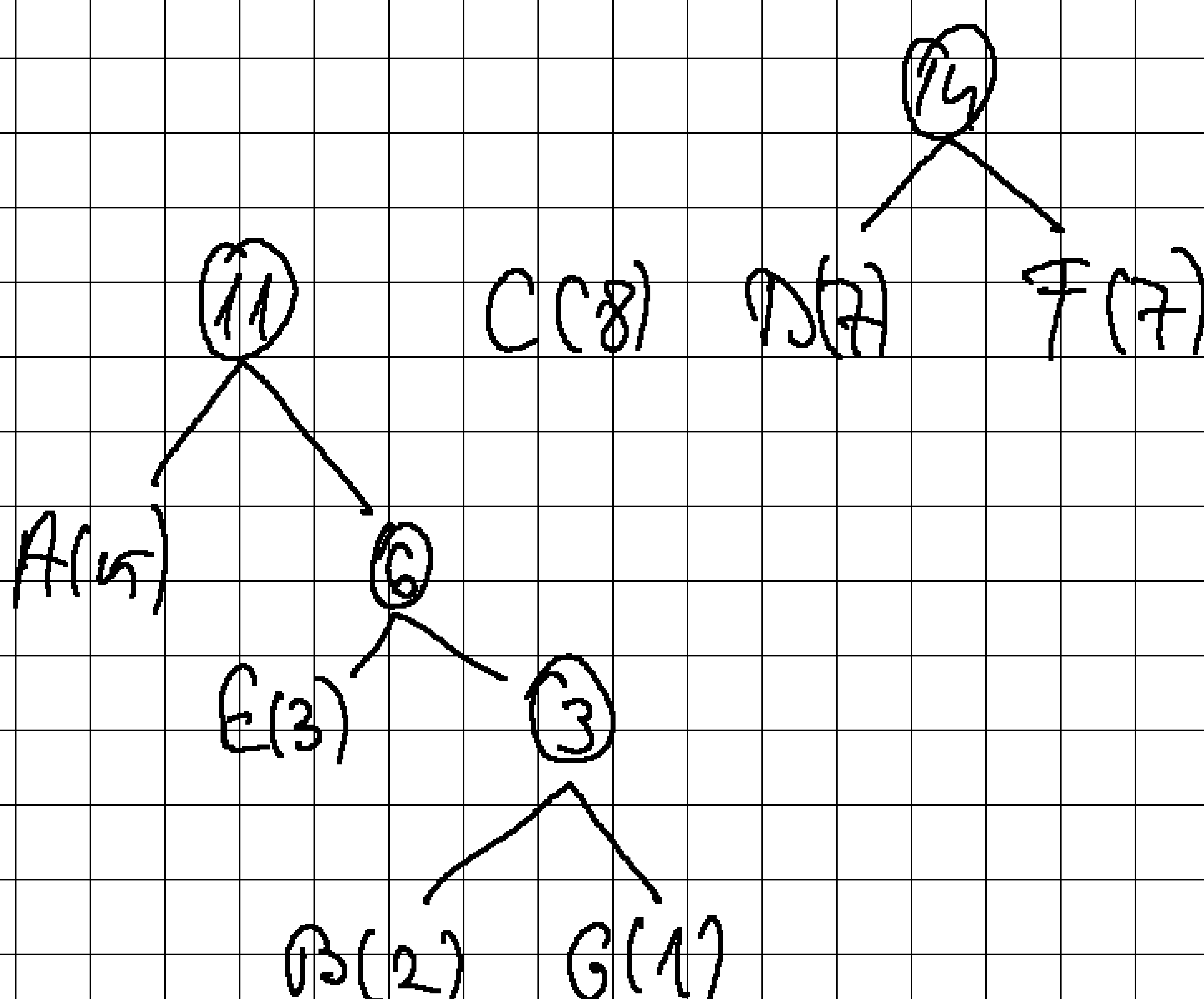
$N(1) = 2$ $N(3) = N(2) + N(1) + 1 = 7$

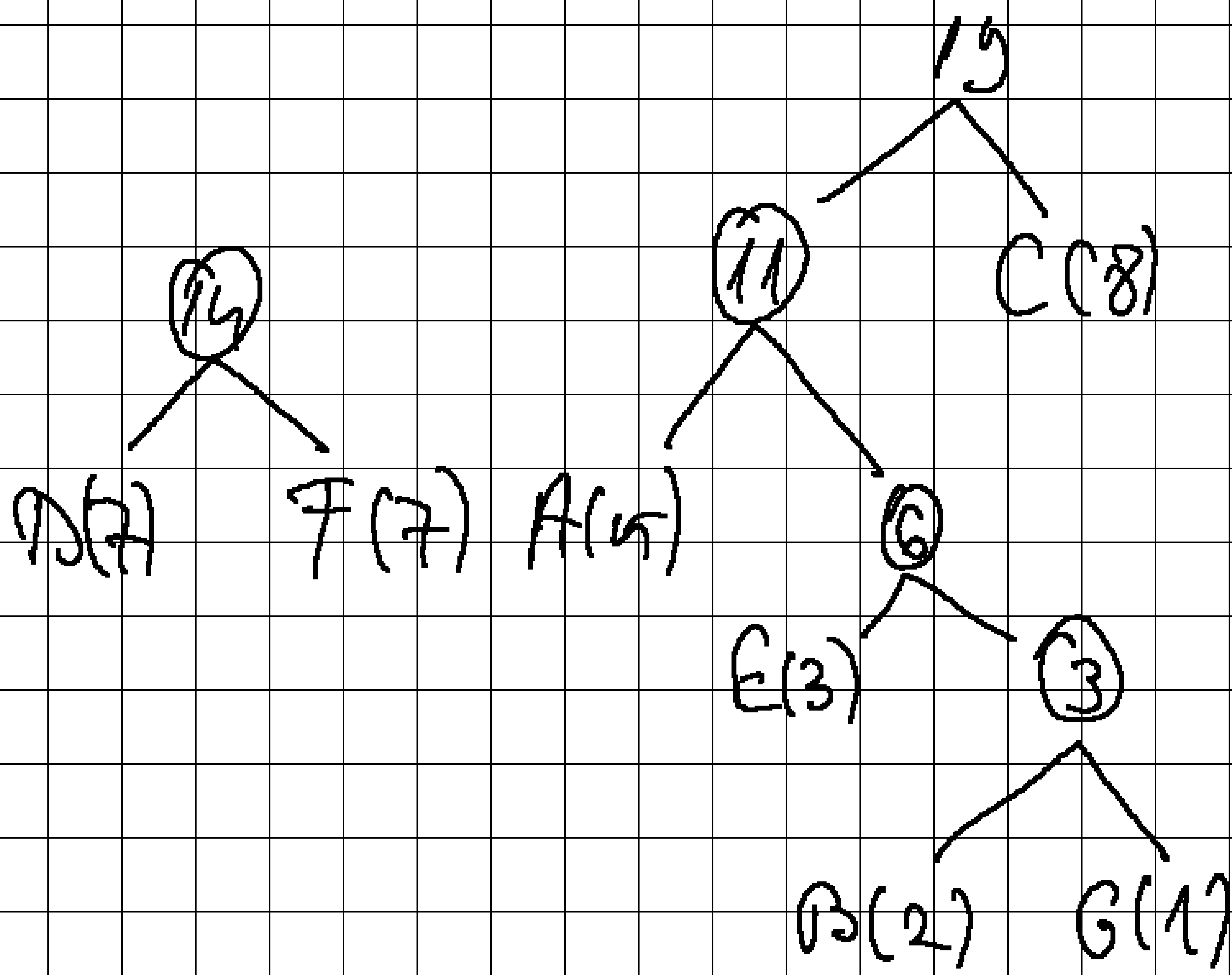
$N(2) = N(1) + N(0) + 1 = 4$

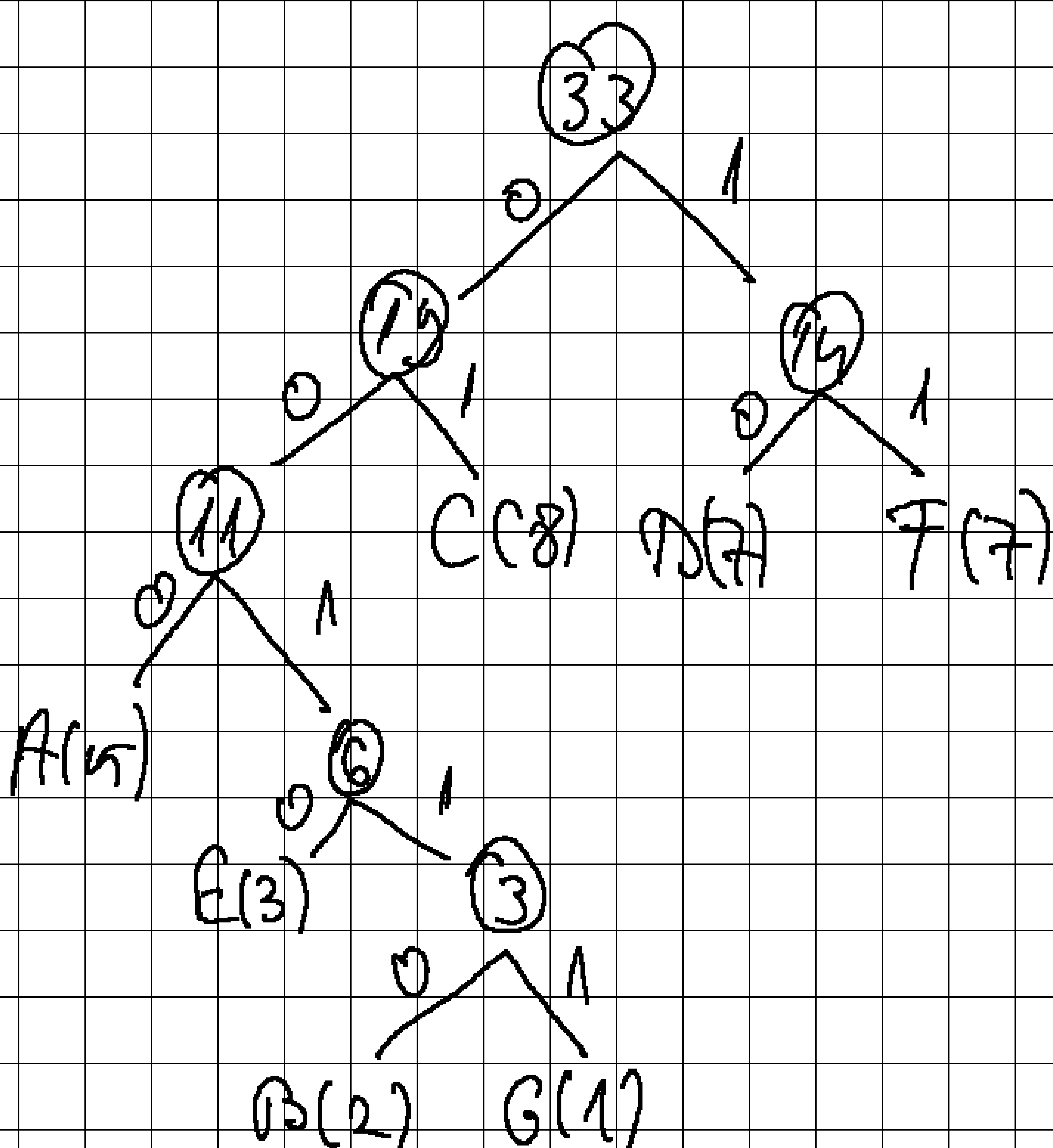

3
C(8) D(7) F(7) A(5) E(3) B(2) G(1)


6
3
C(8) D(7) F(7) A(5) E(3)
B(2) G(1)


11
C(8) D(7) F(7) A(5)
6
E(3)
3
B(2) G(1)


14
11
C(8) D(7) F(7)
A(5)
6
E(3)
3
B(2) G(1)





F: 11
A 000
C 01
E 0010

FACE: 11 000 01 0010

Length 11: