SPLAY Tree

What is splay tree?

Your paragraph text

A Self adjusting binary search tree.

Why we learn splay tree?

For the advantage of the quick read and access the element.

Caching systems

Dynamic Search Tree

Network routing algorithom

Database systems

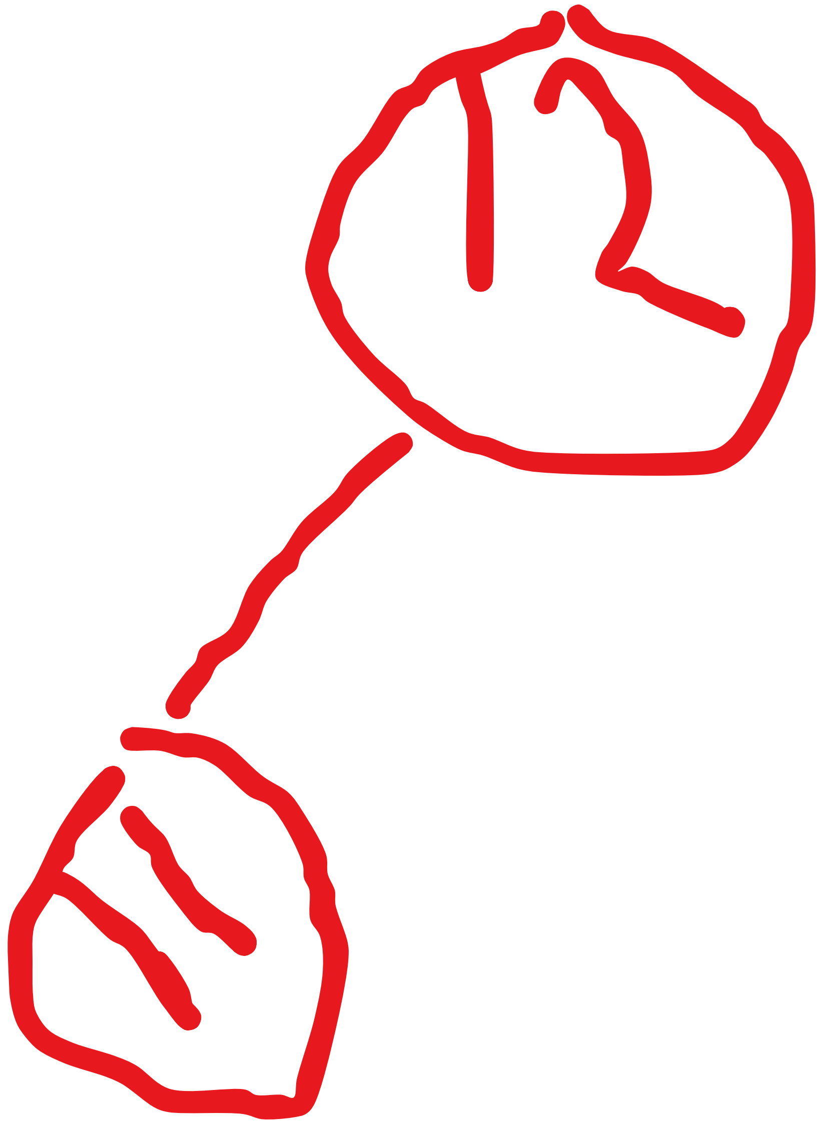
Text Editors and data compressions

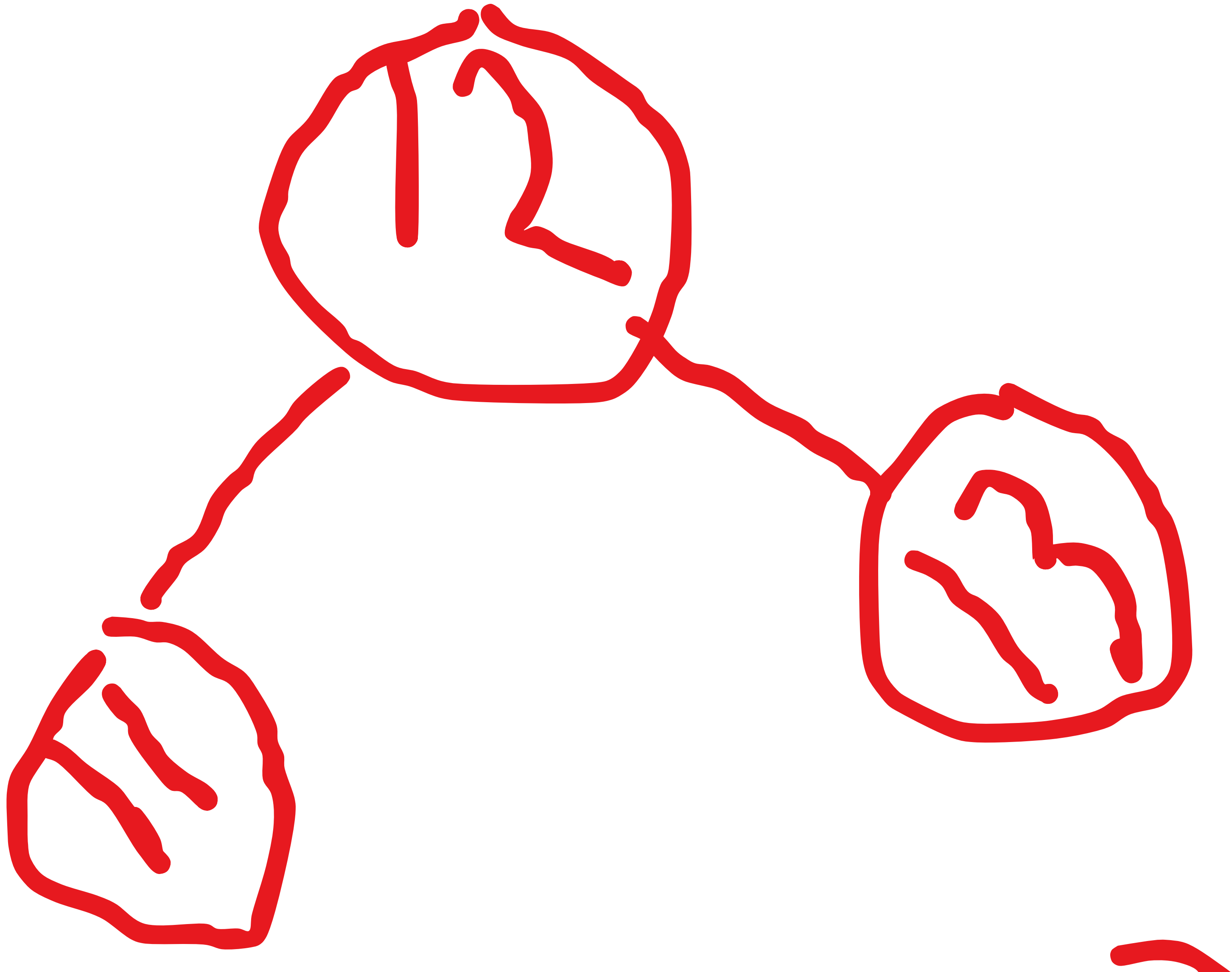
Online algorithom optimizations

File systems

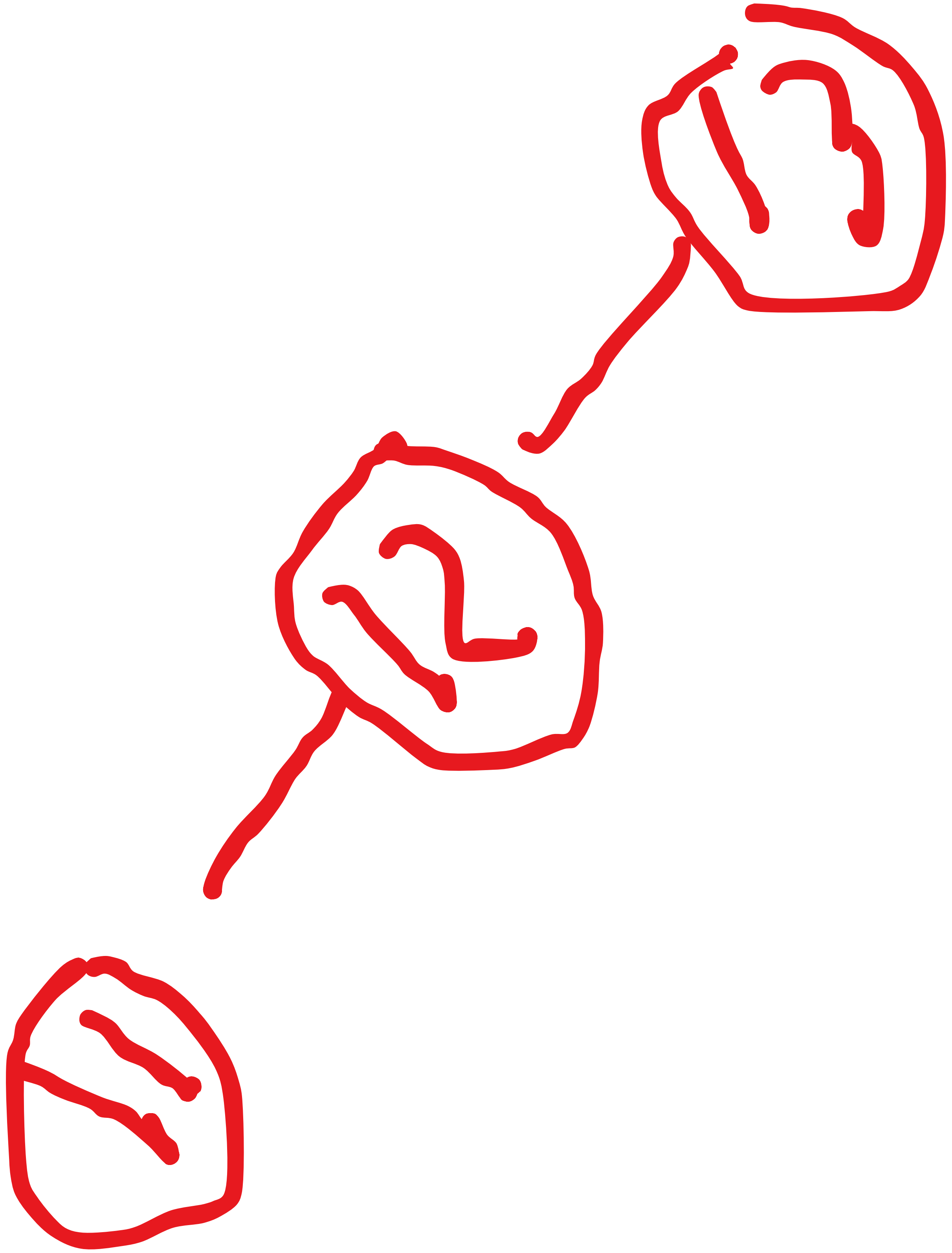
How it adjust the tree?

this tree takes all recently updated element in root.





add 13 at
here



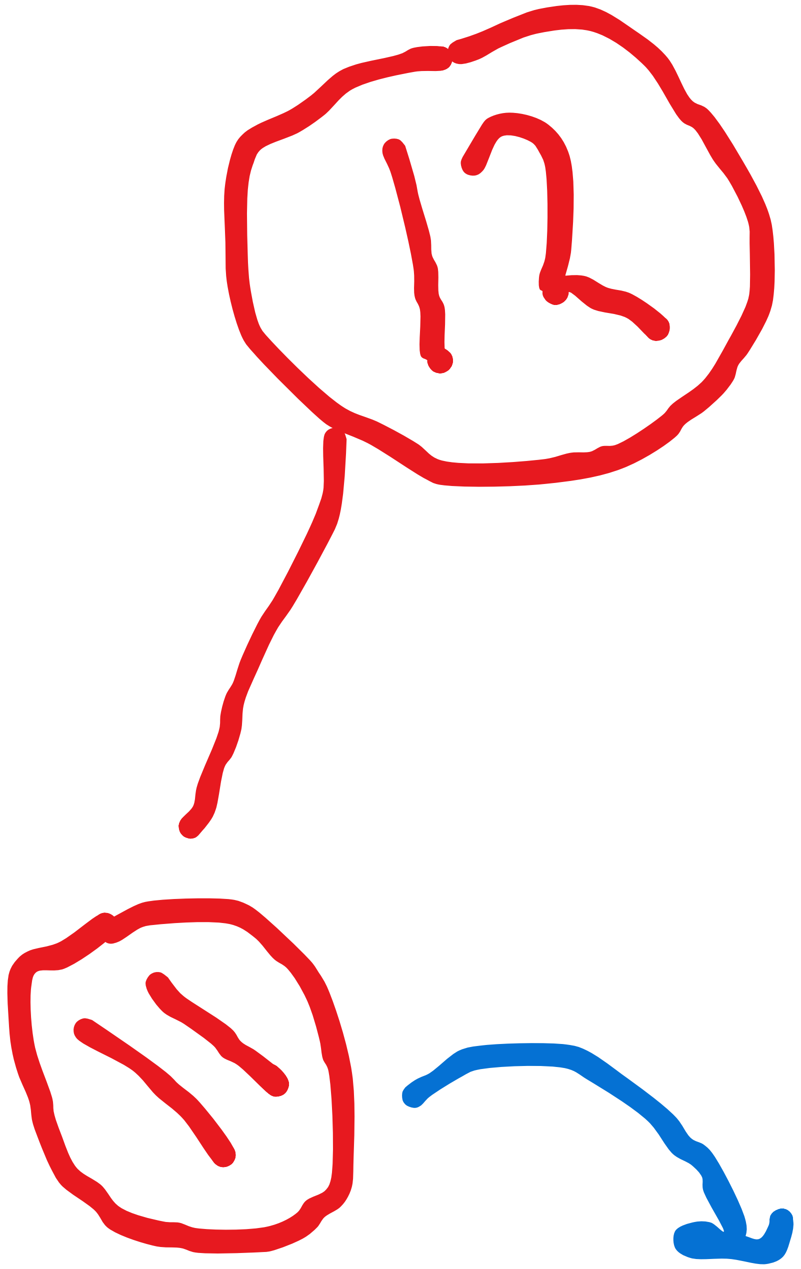
after addition

Let’s solve all of its issue.

Its addition deletion and searching is same as red black tree

It also adjust the tree using the rotations.

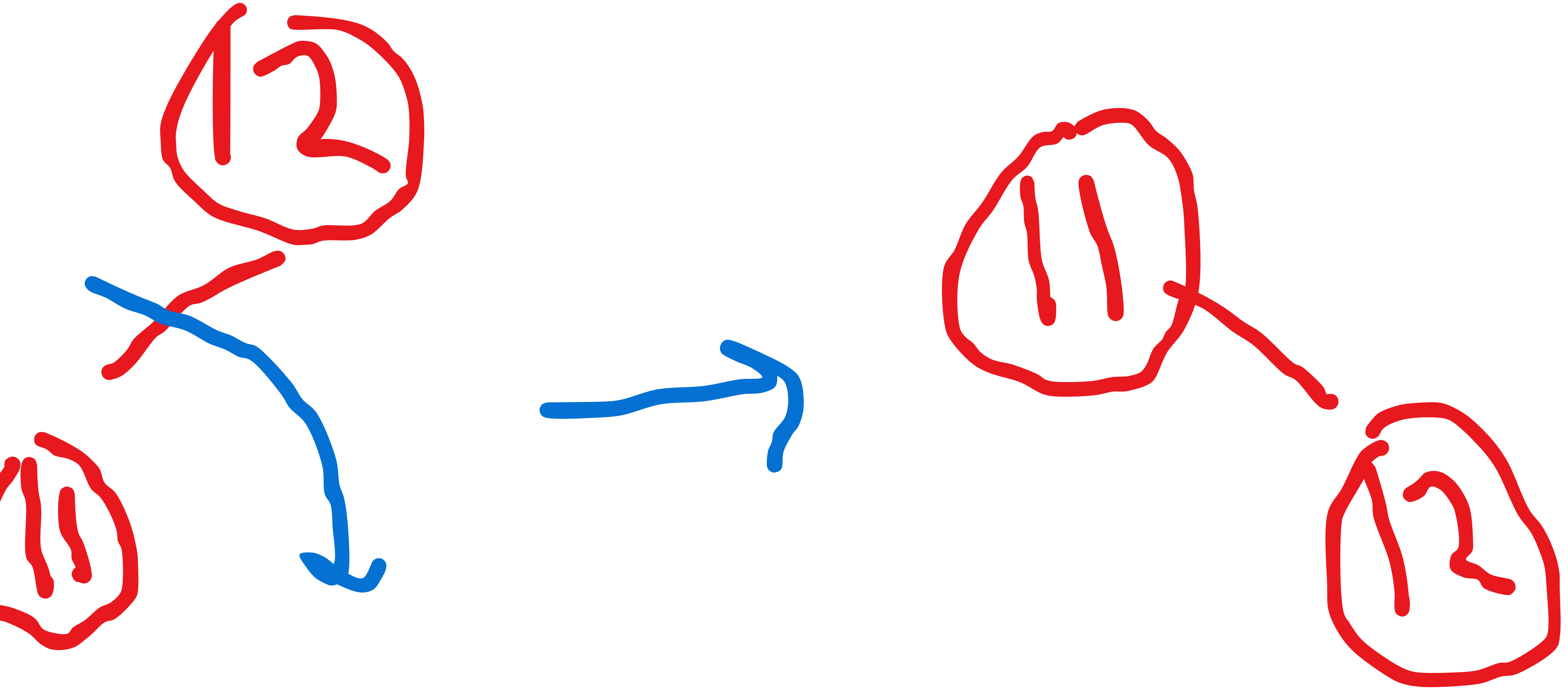
- left rotation
- right rotation

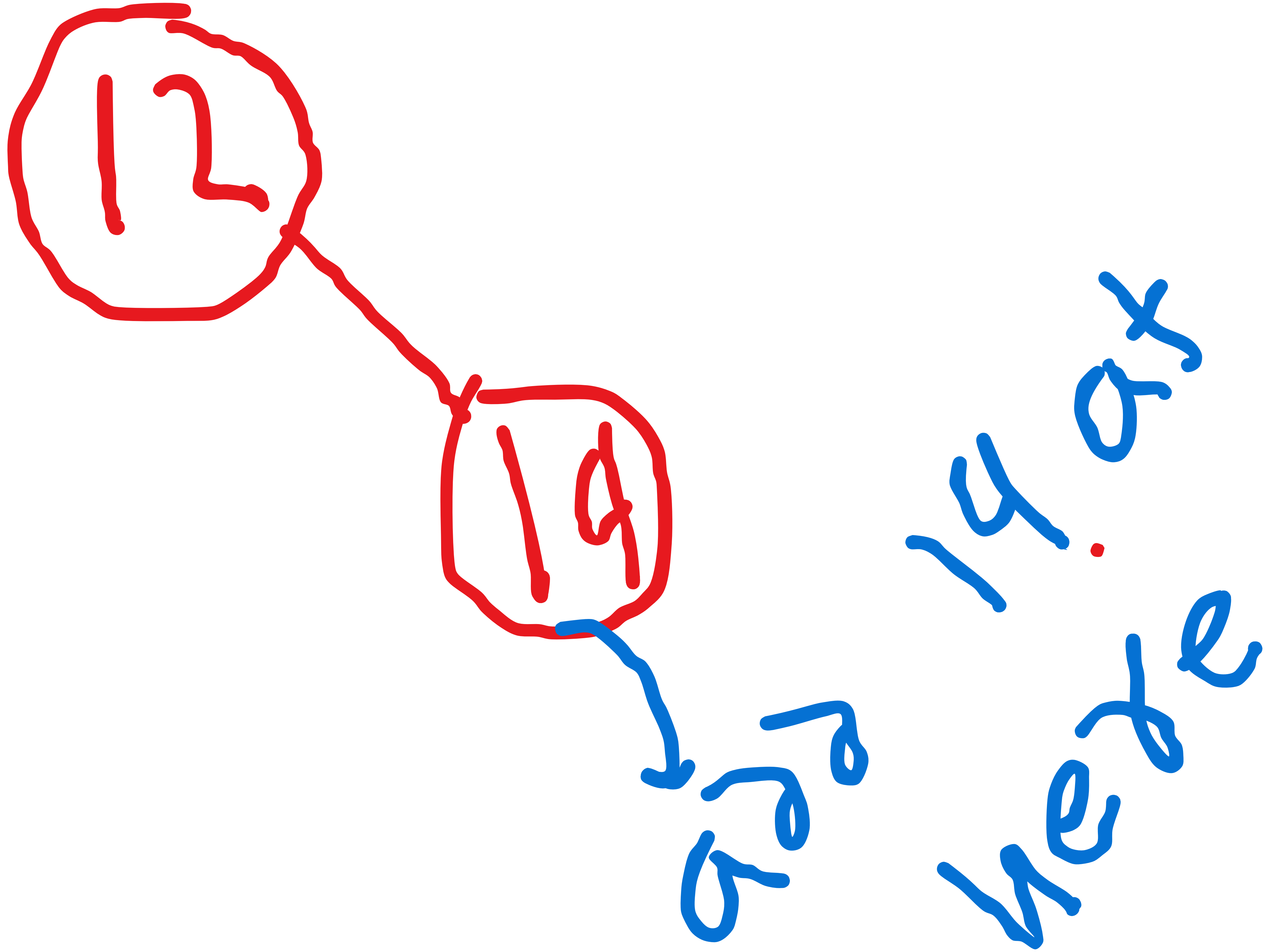


11 is added at
here

Left heavy sitution / zig sitution.

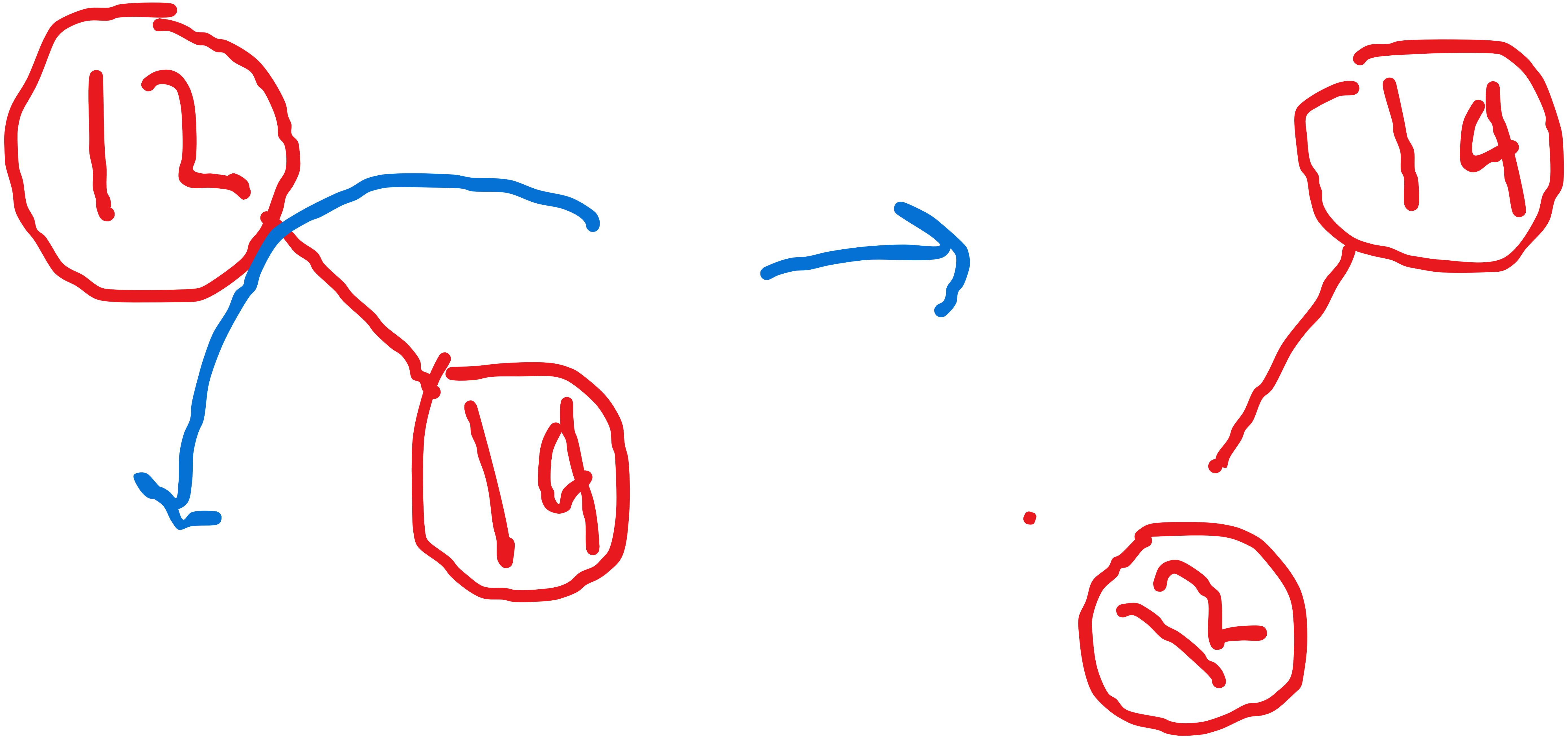
Here we perform the right rotations.

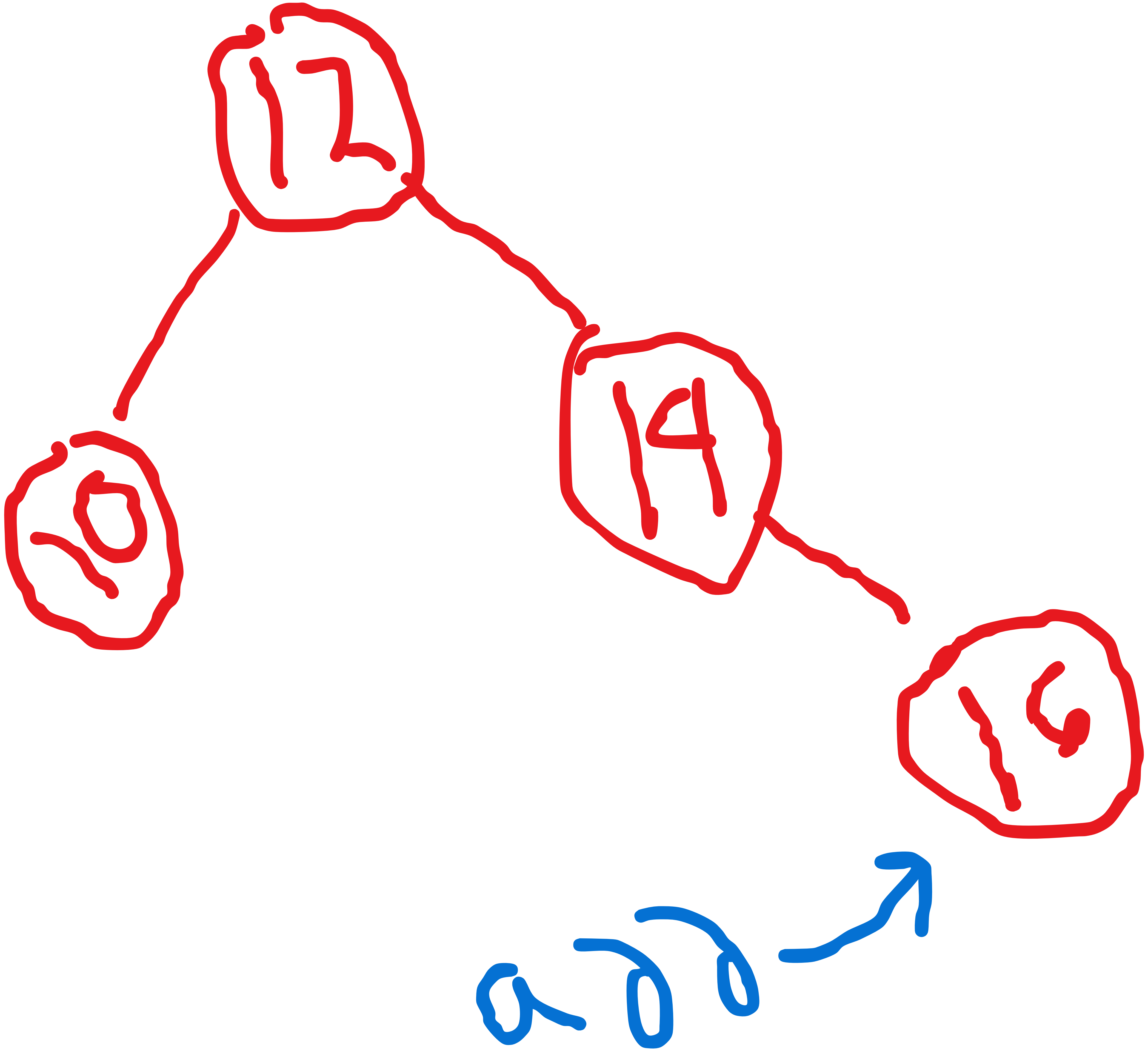

12
11
11
12

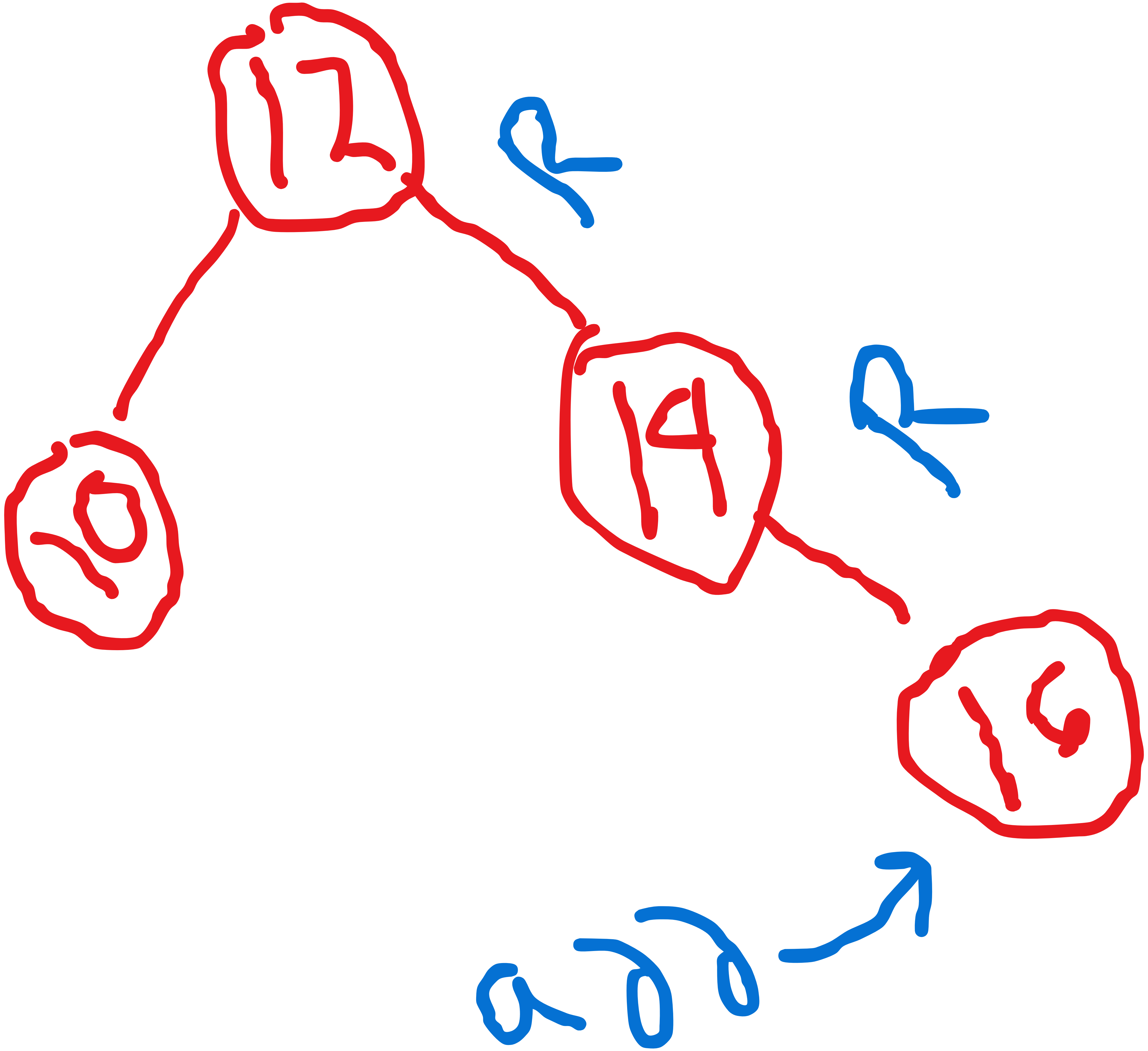

12
14
add 14 at here

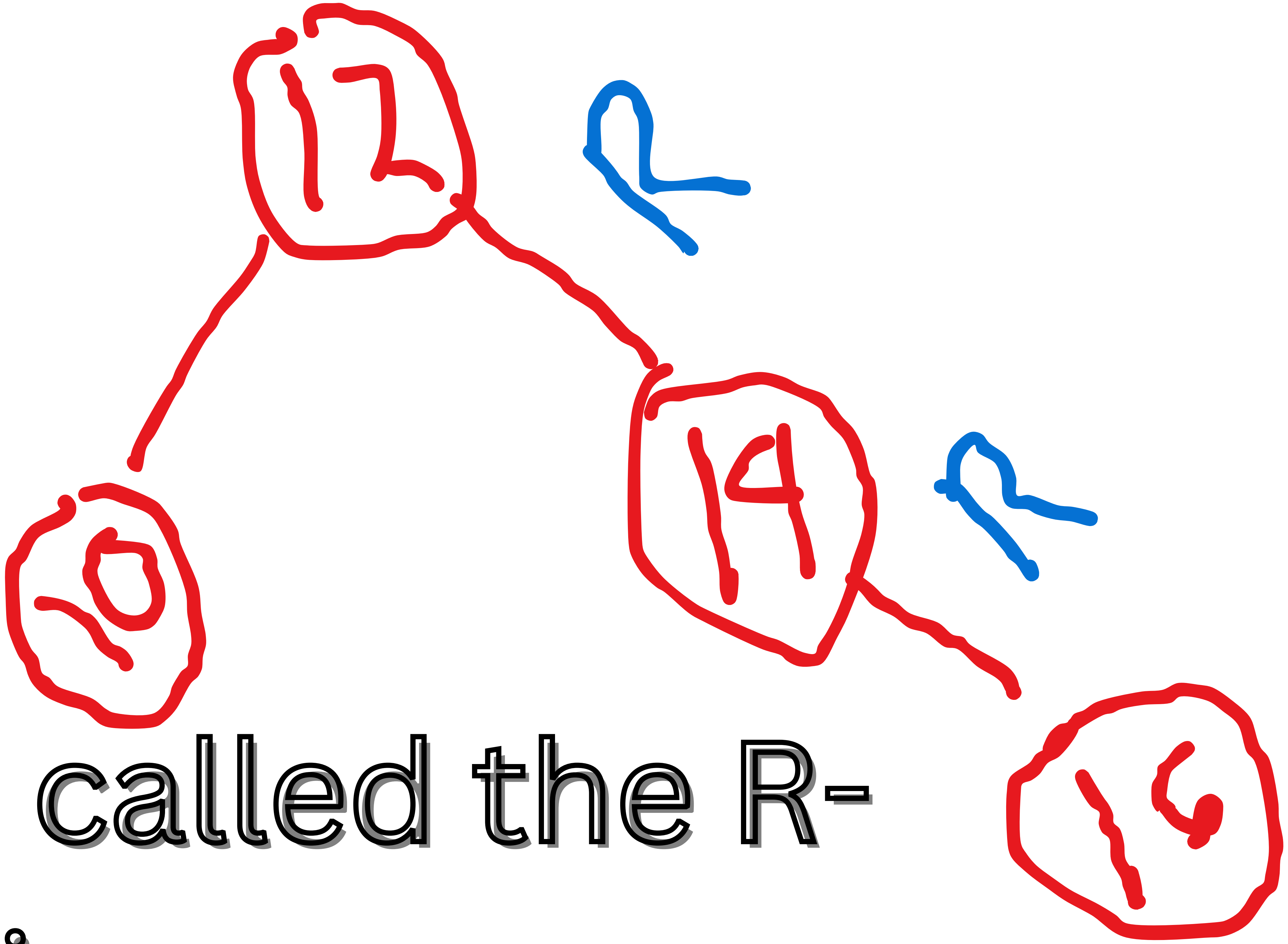
right heavy / zag sitution

Here we need to perform the left rotations.


12
14
14
12

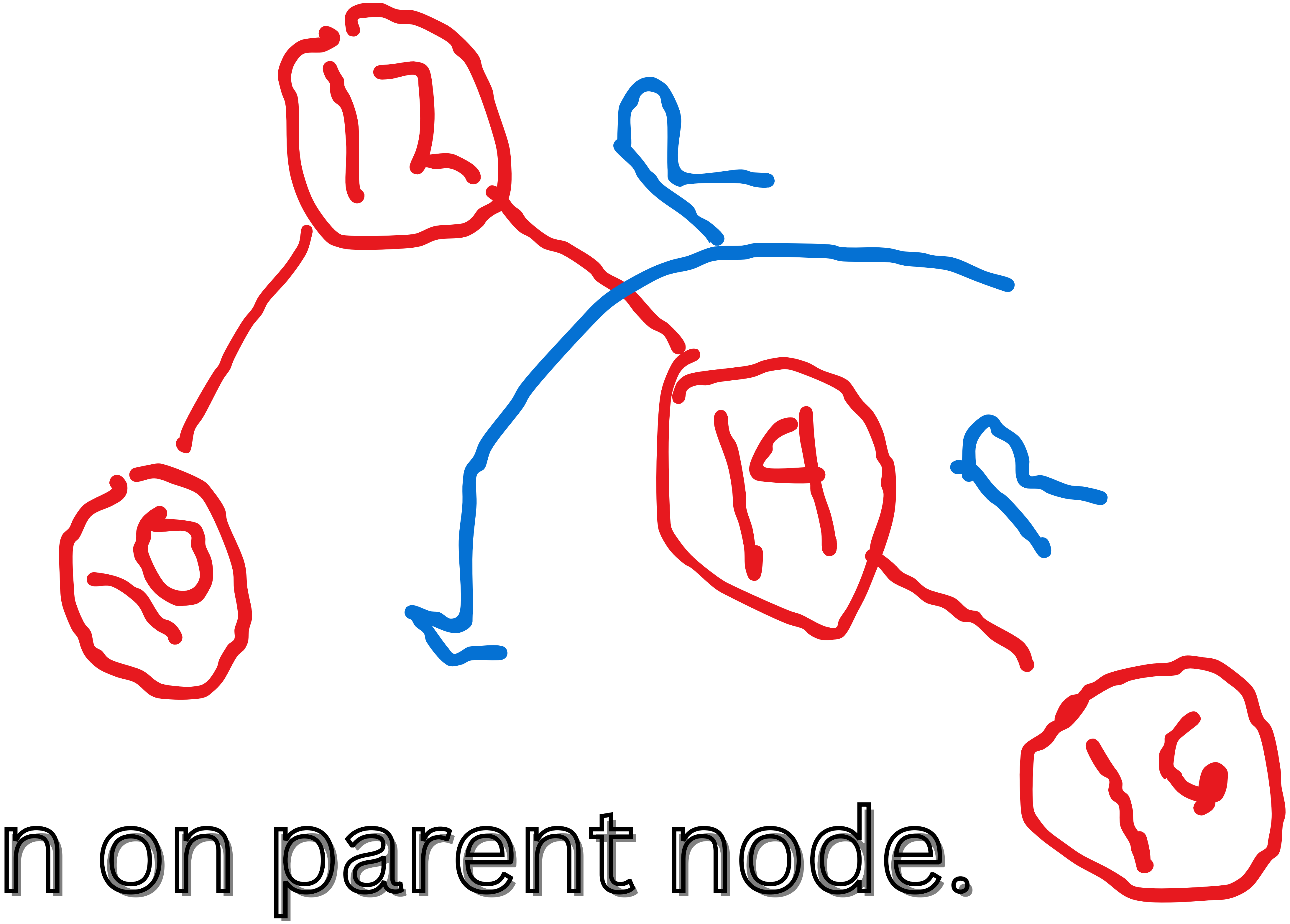

12
10
14
16
add


12
10
14
16
R
R
add



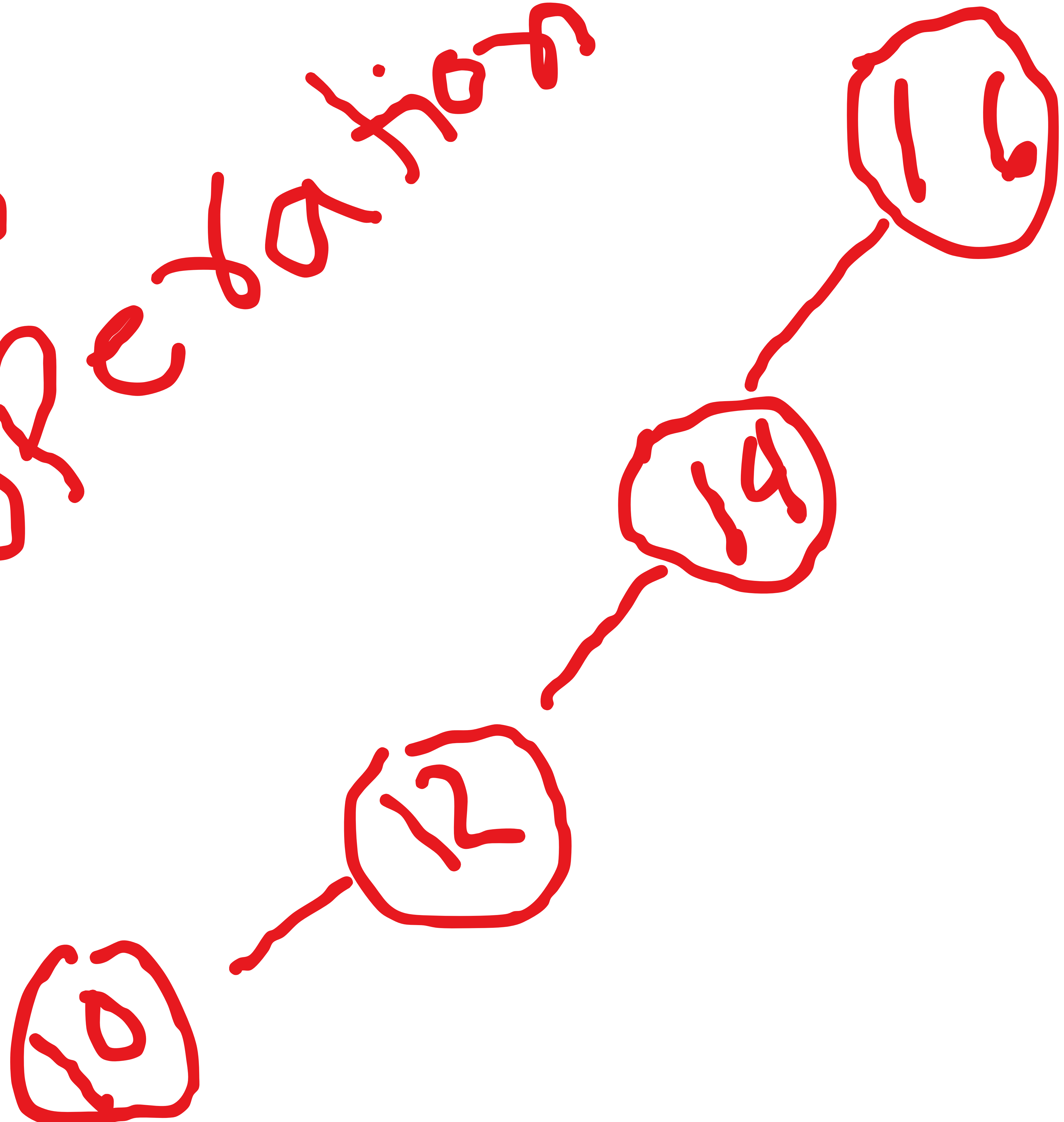
This is called the R-R sitution or zag zag sitution

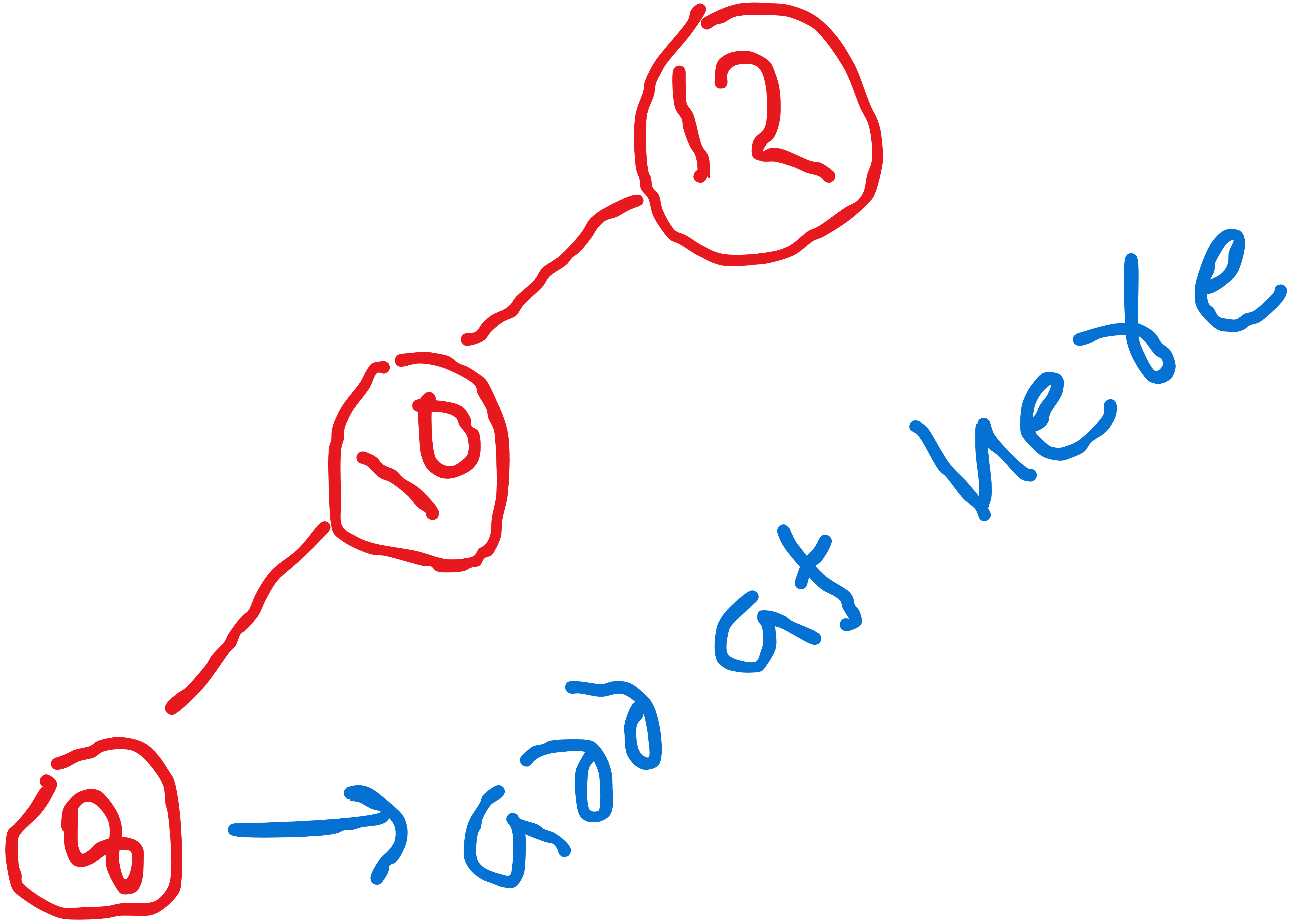
Here we need to perform the left rotation

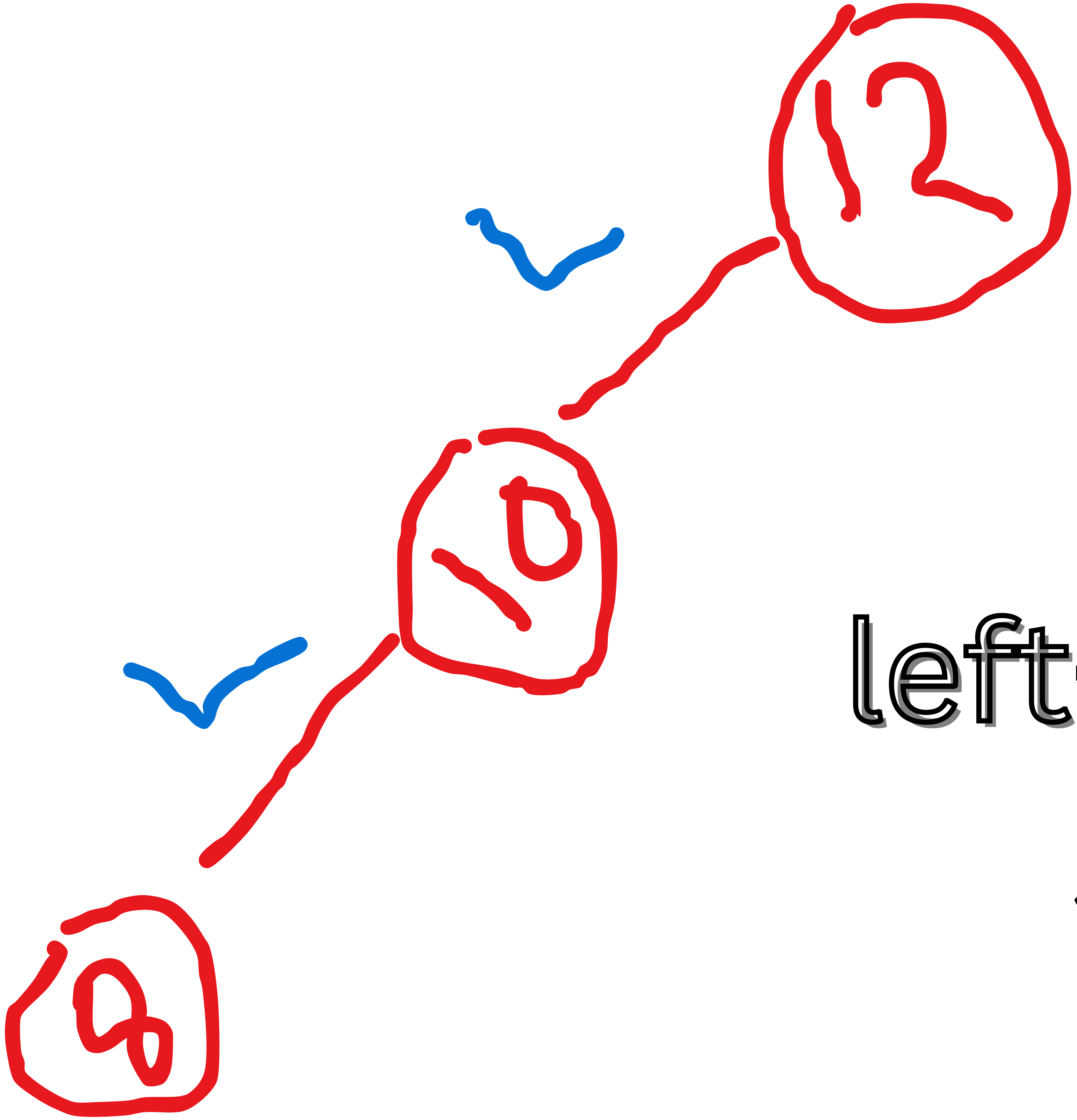


left rotation on parent node.

after operation

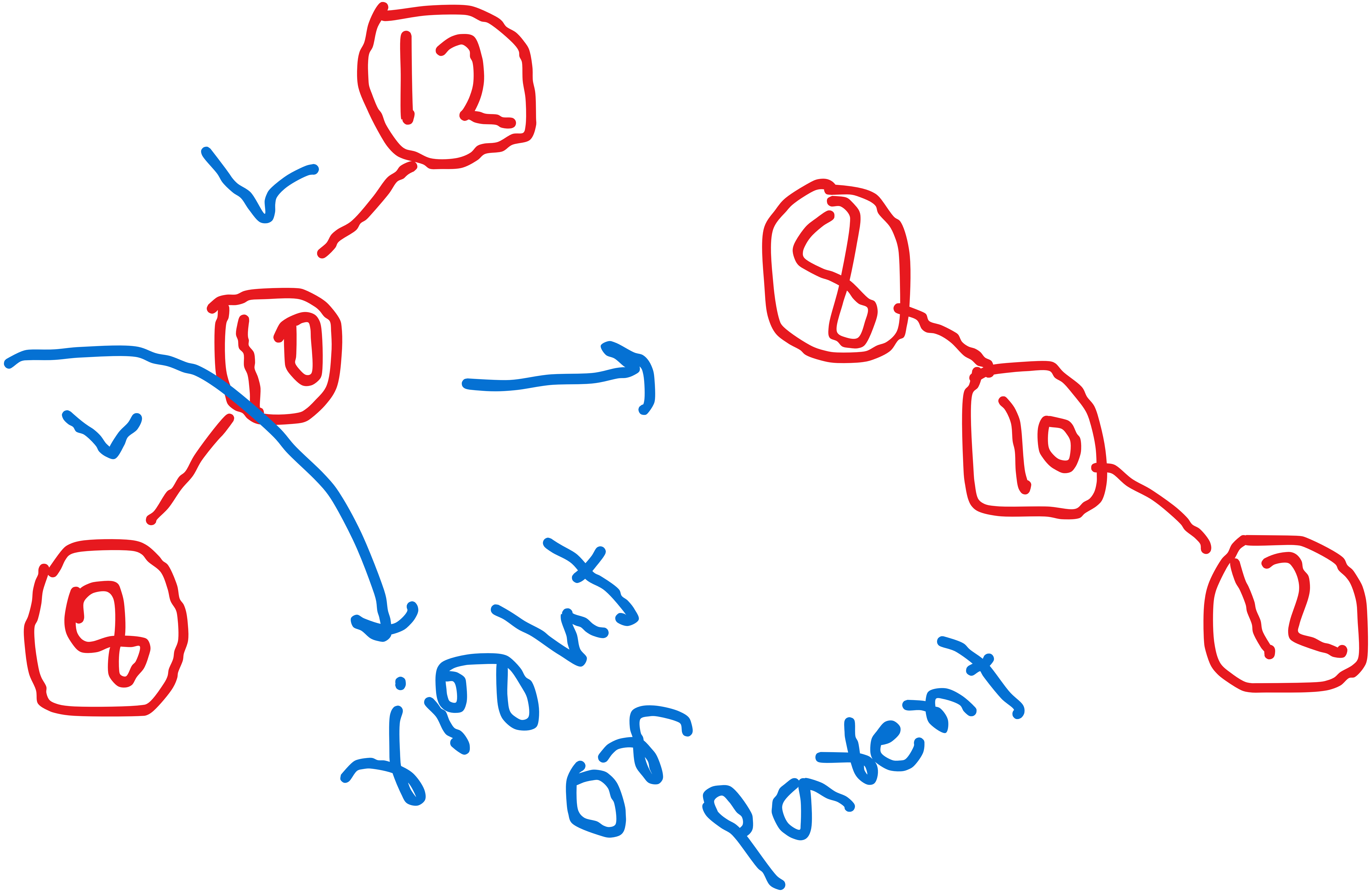


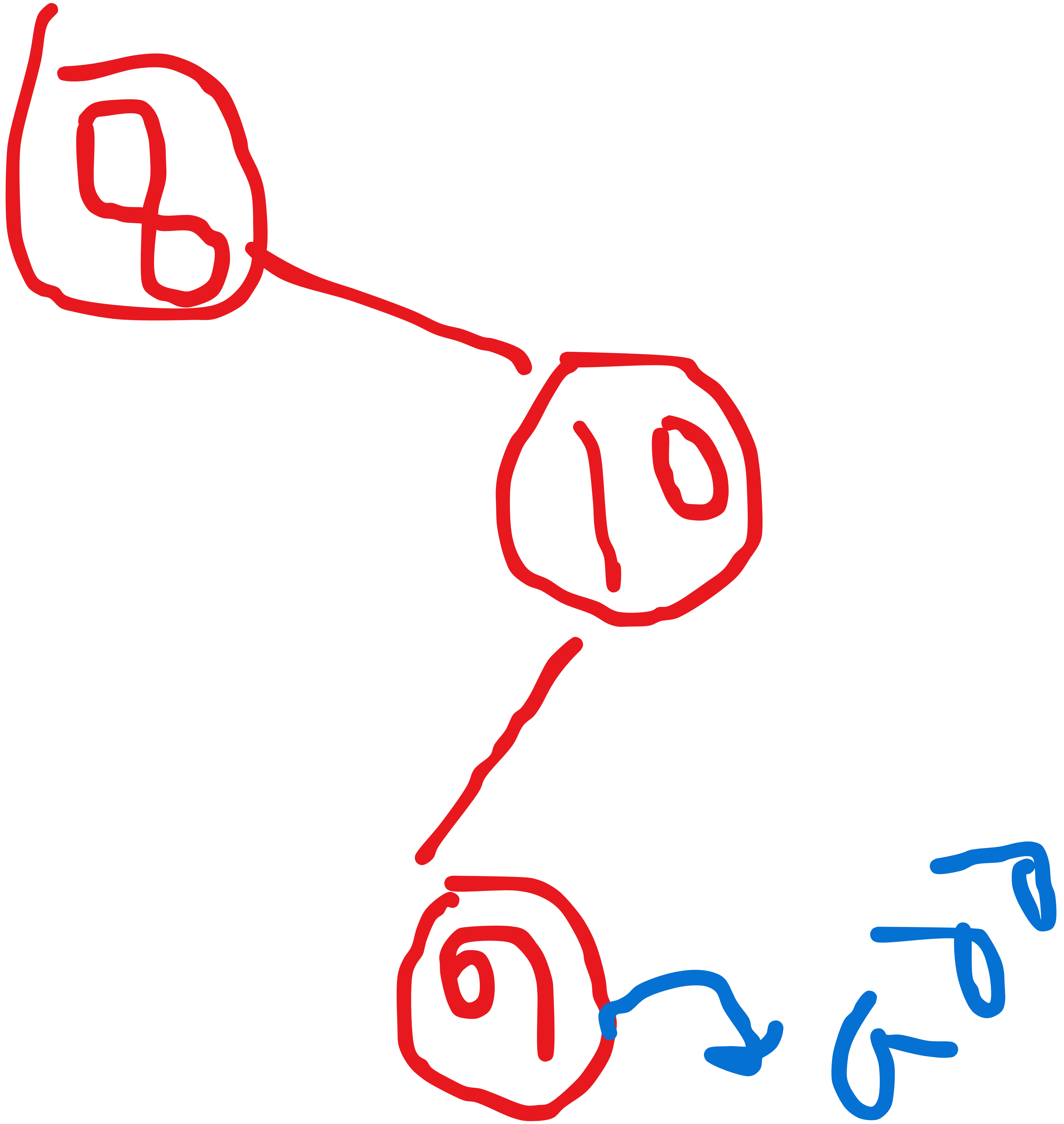

12
10
8
add at here

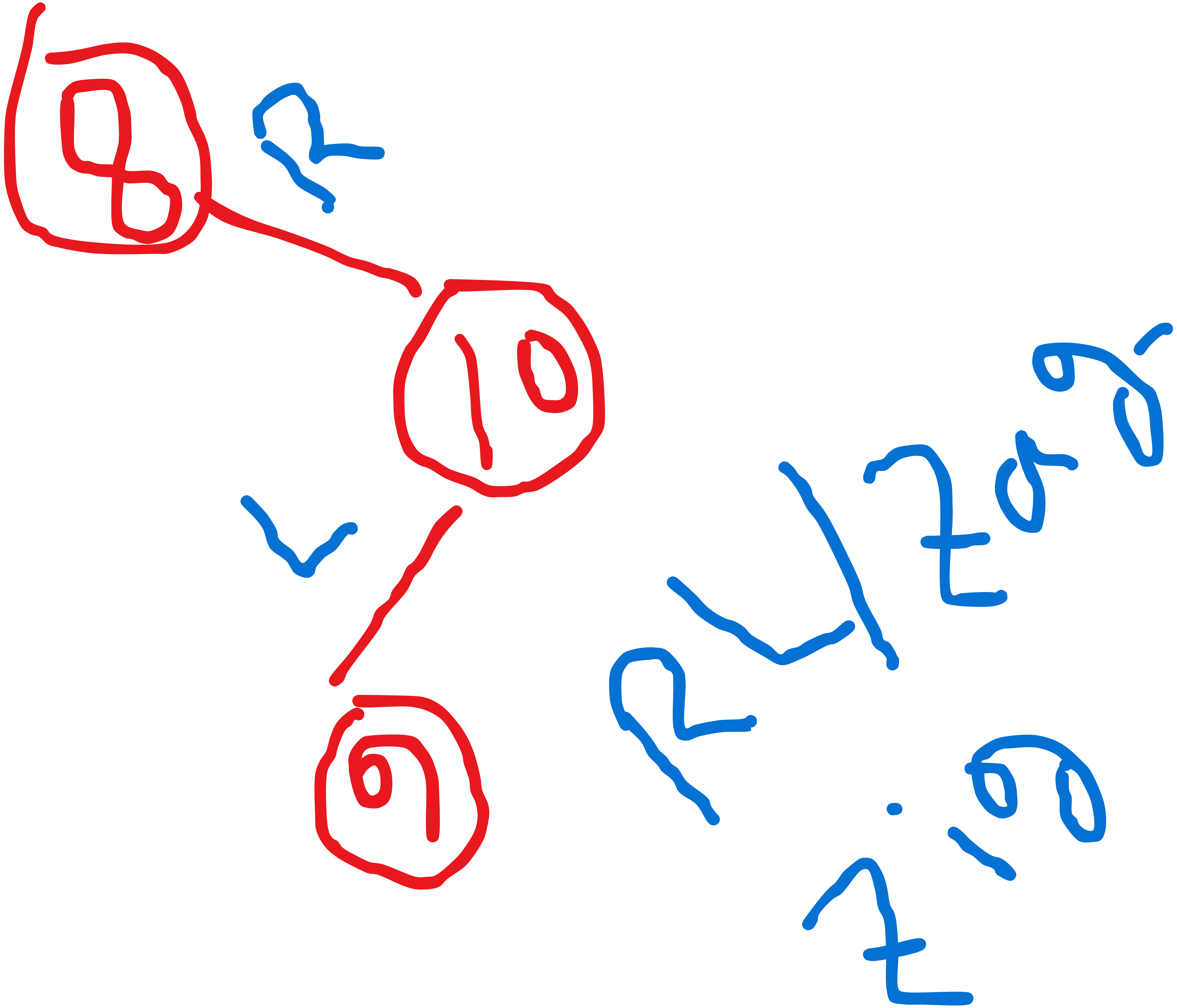


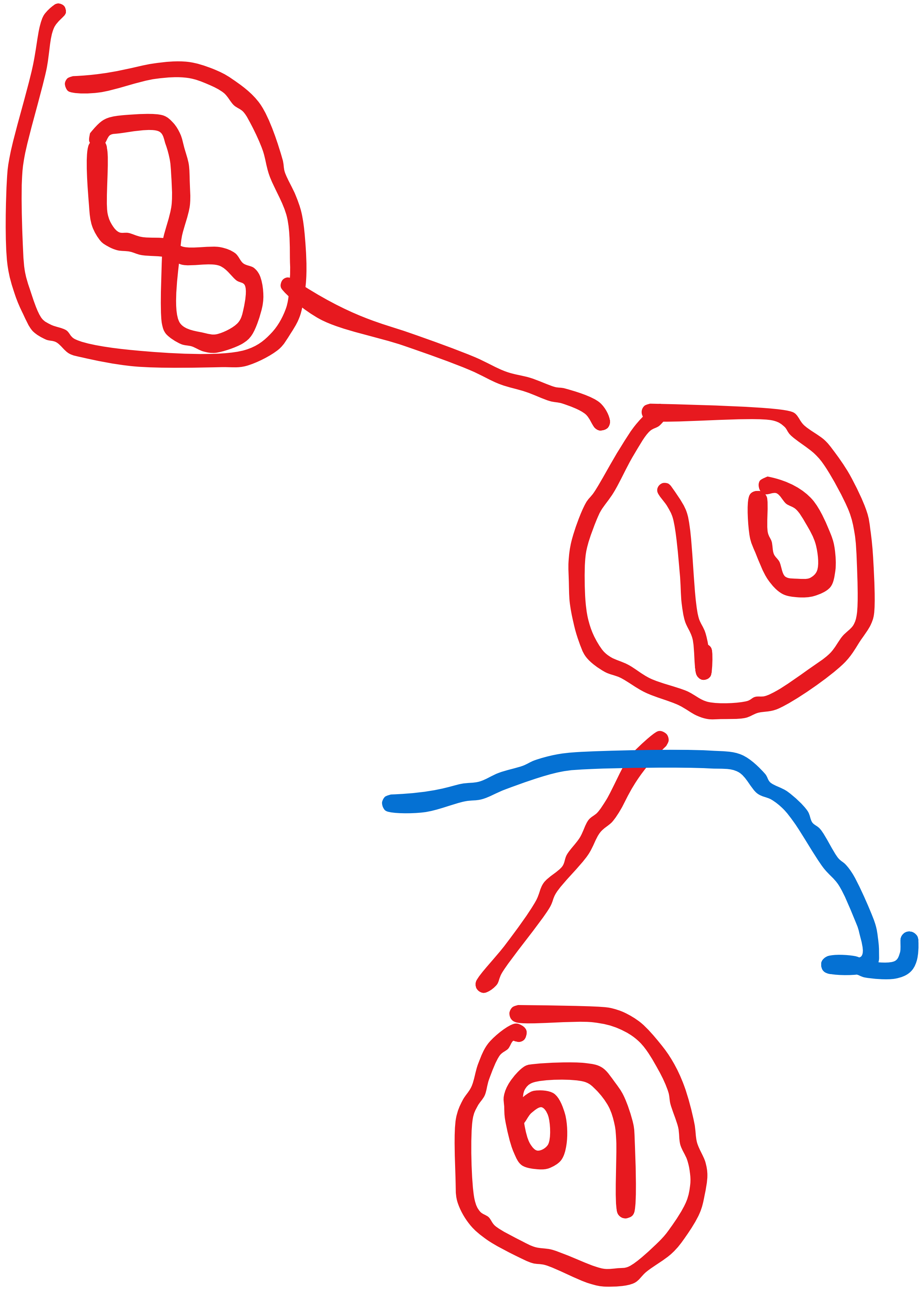
left-left / zig-zig
situtions

Now we perform the right rotations

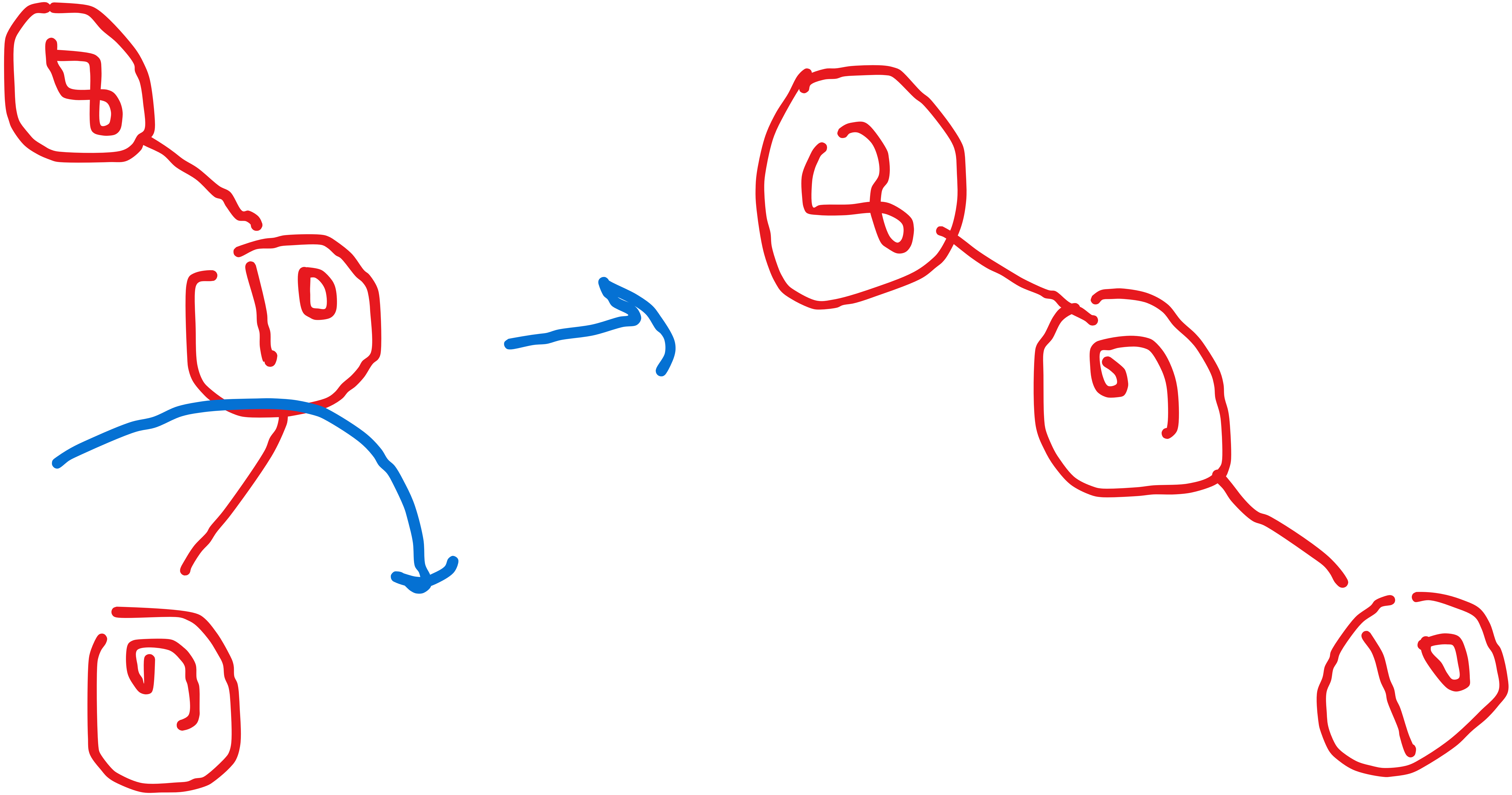

12
10
8
right of parent
8
10
12

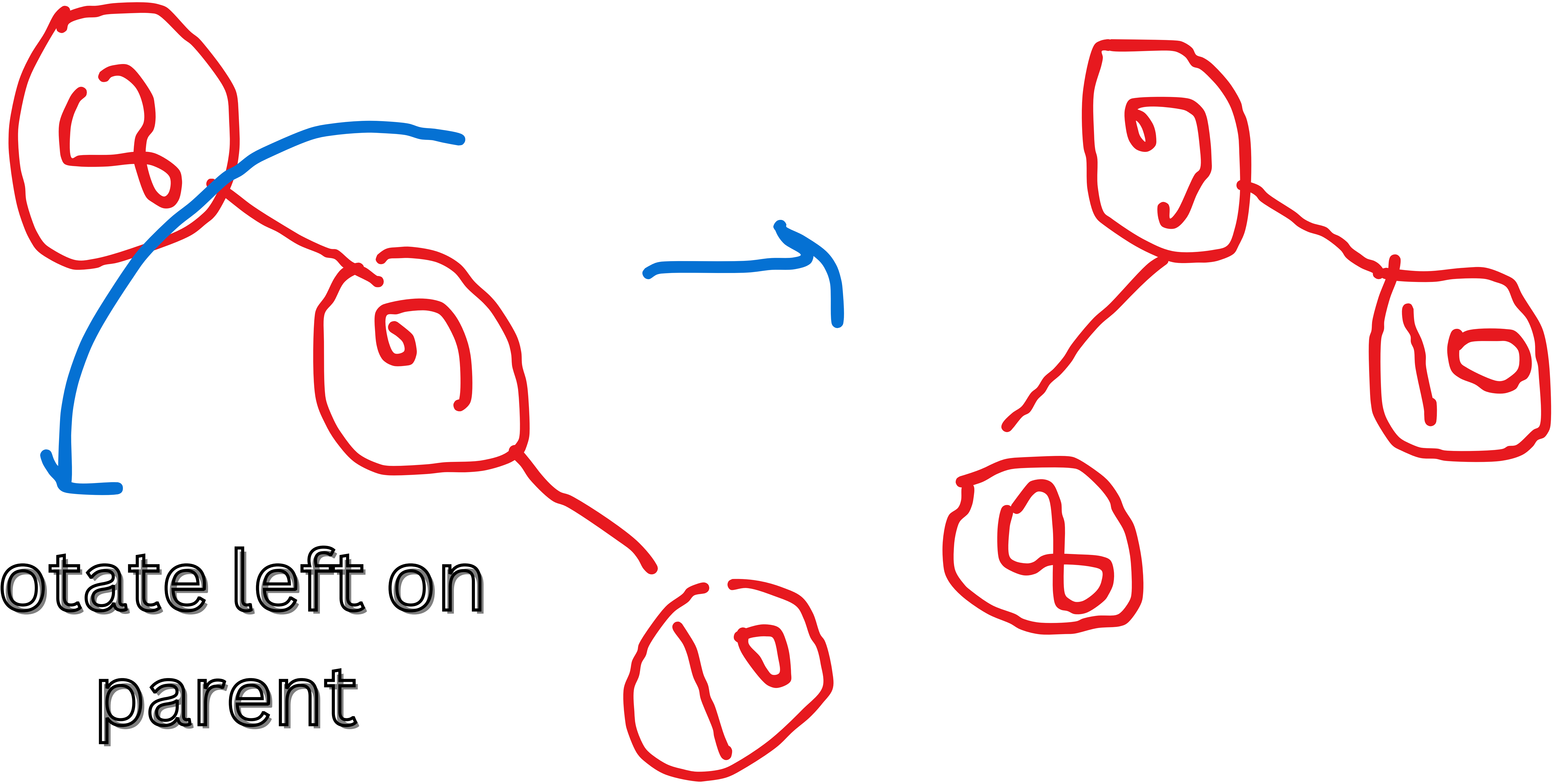

8
10
9


8
R
10
L
9
RL/Zag
Zig

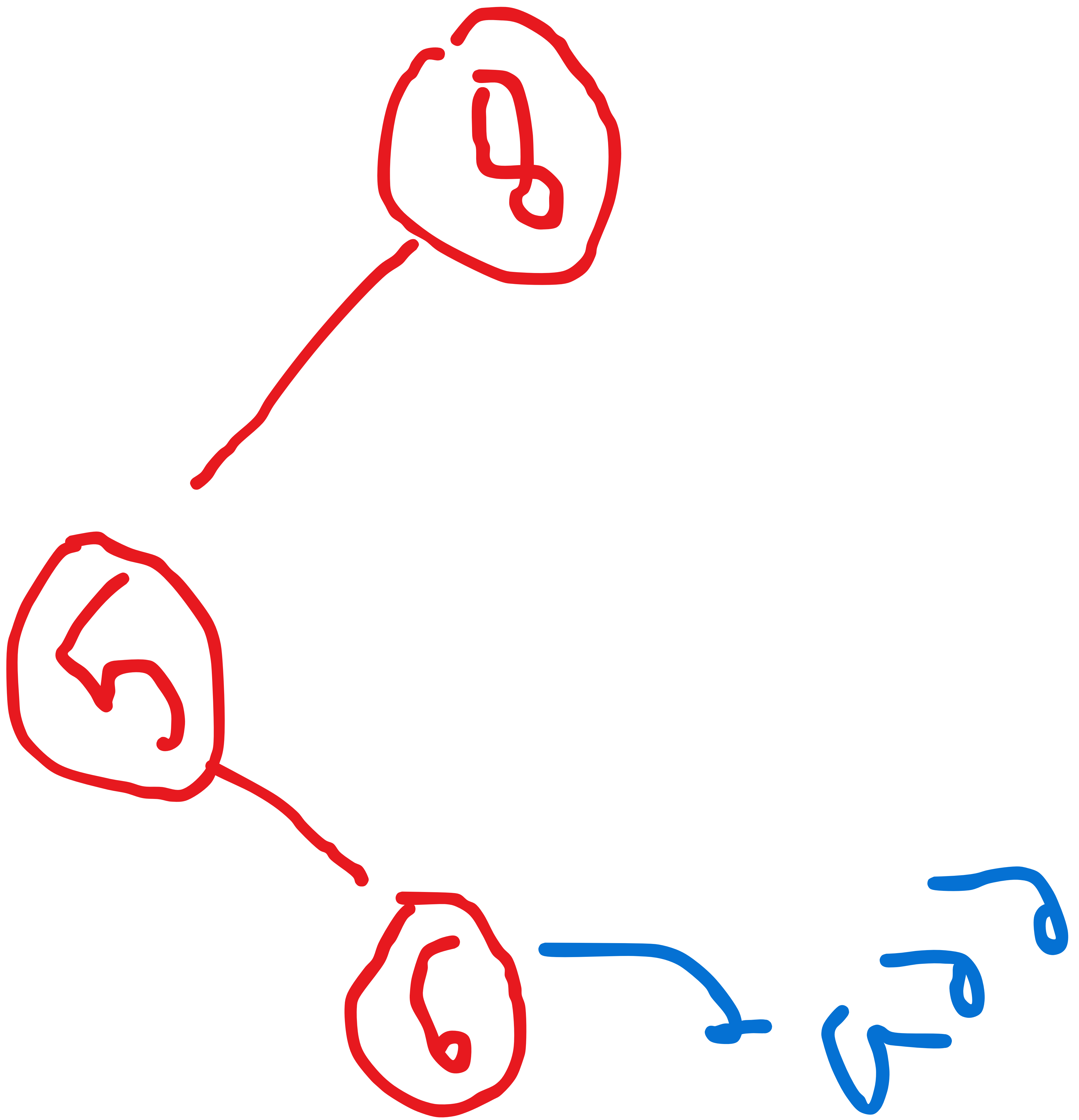


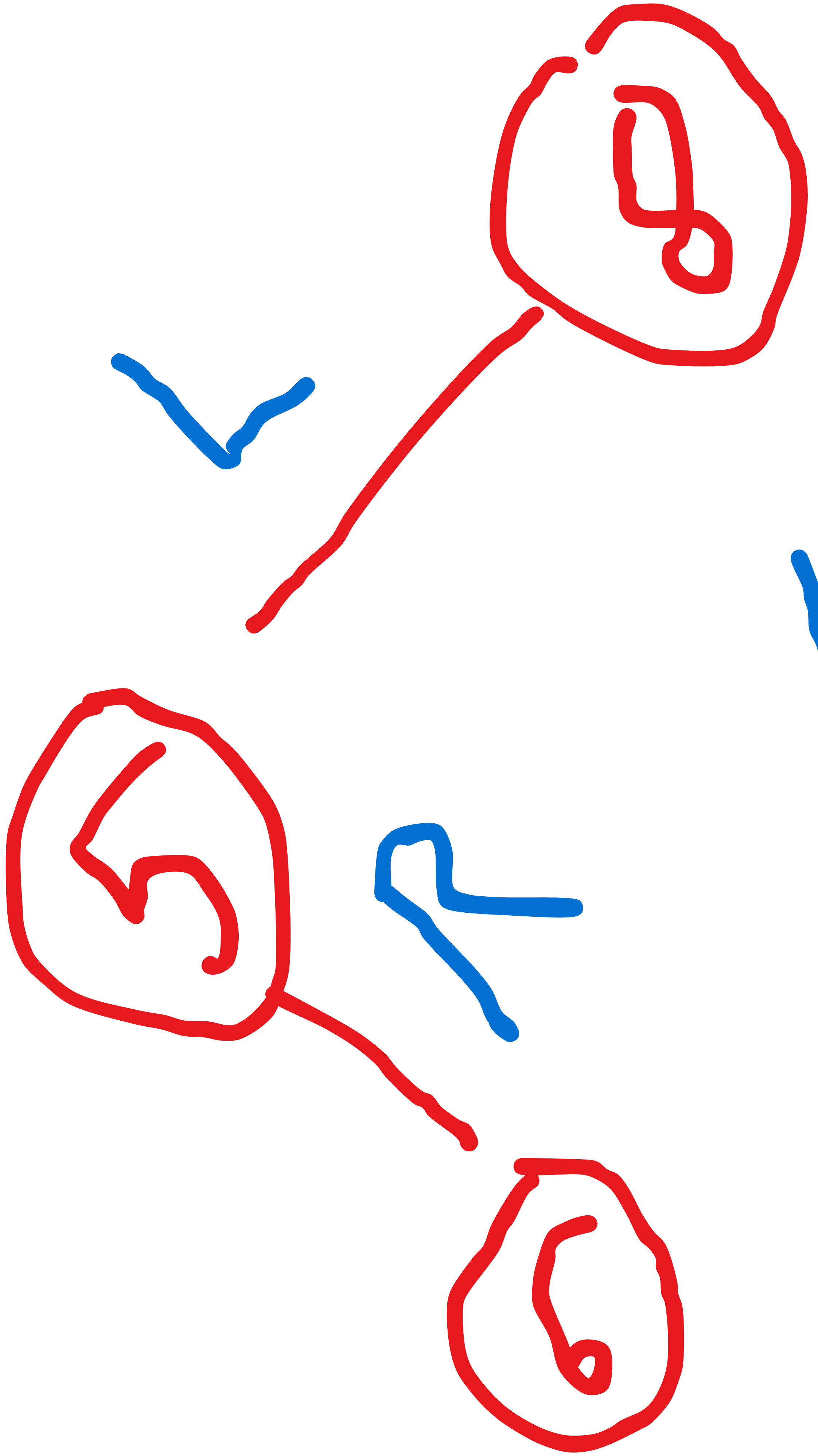
perform right on parent


8
10
9
8
9
10

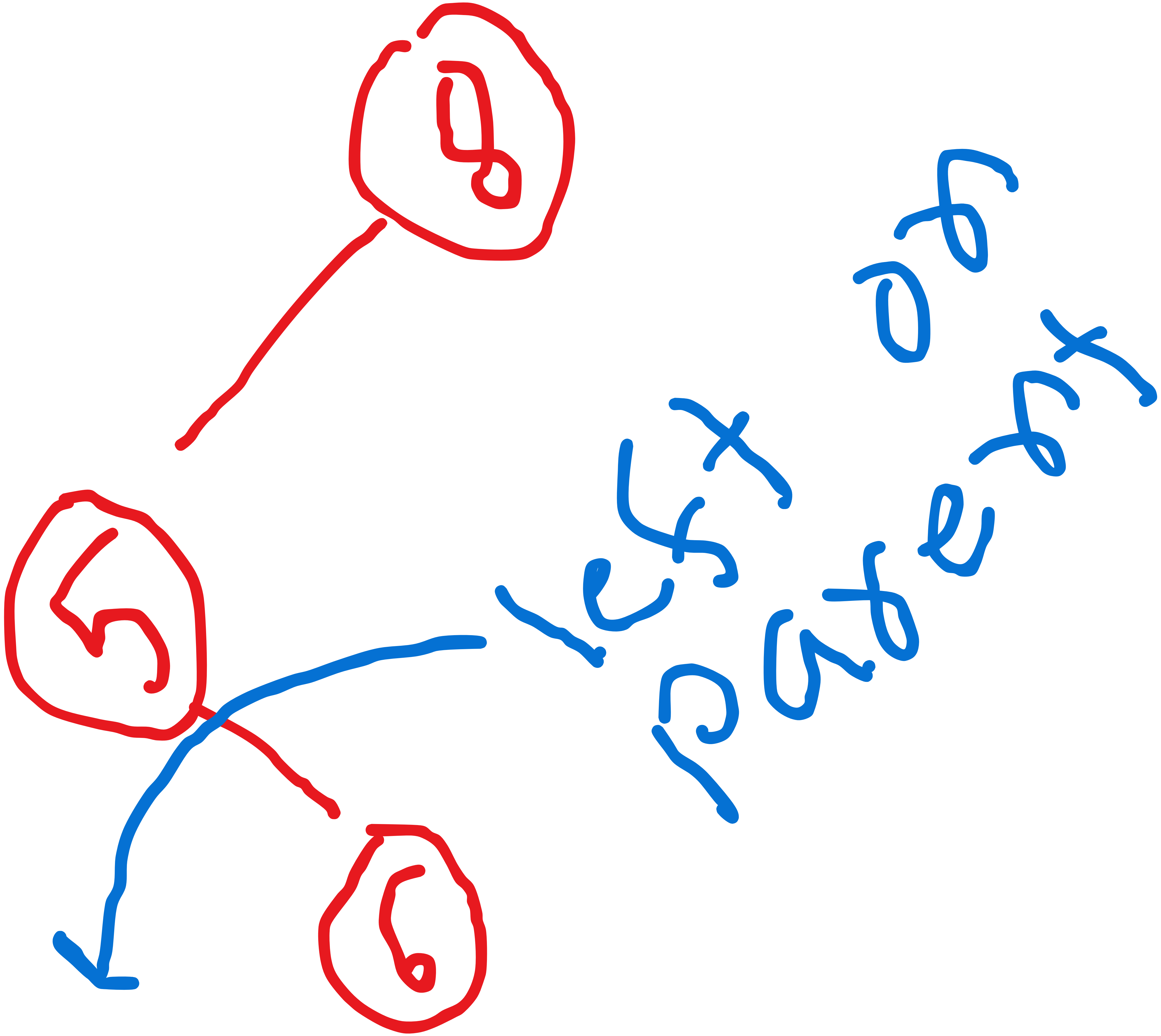


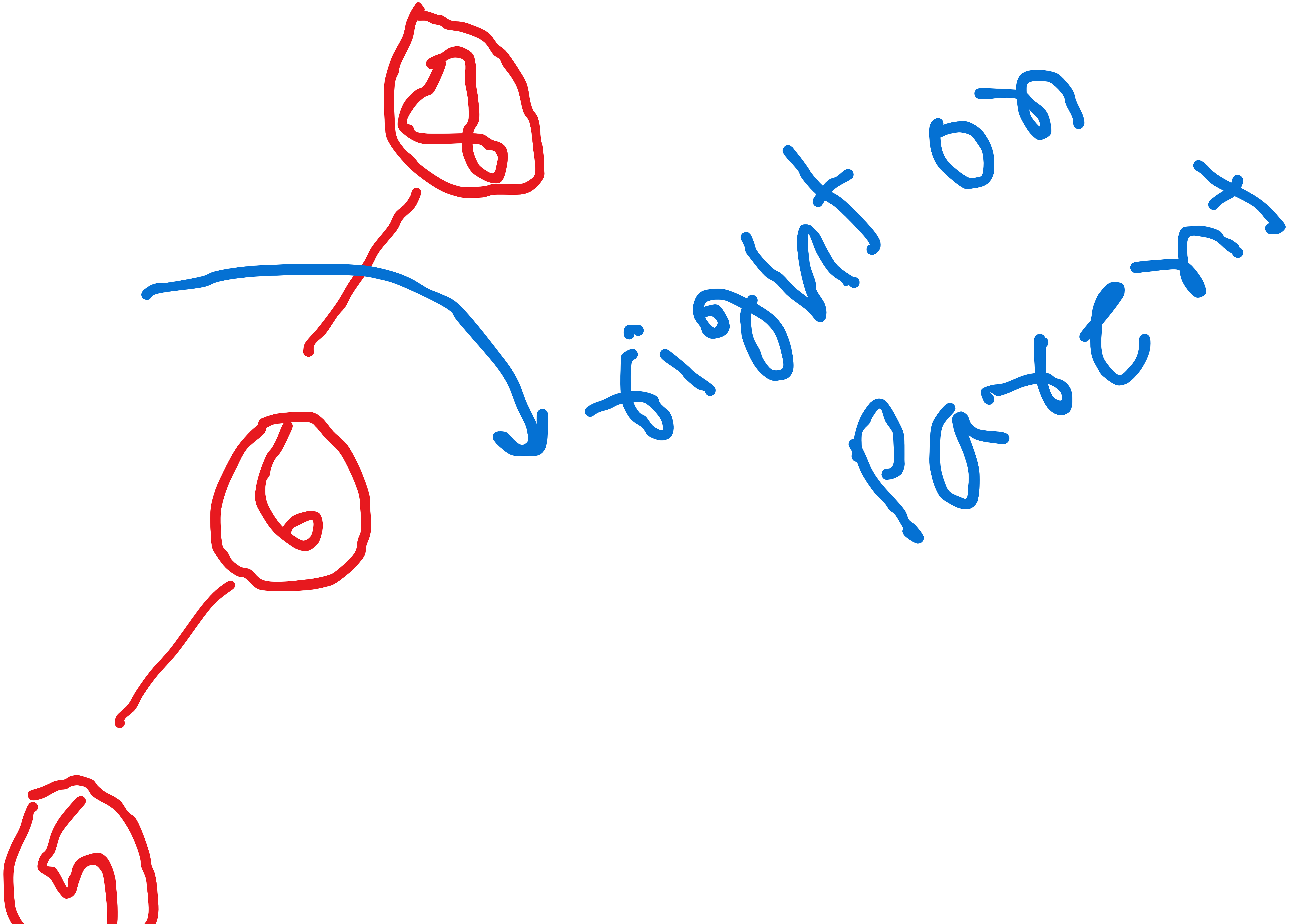
rotate left on
parent

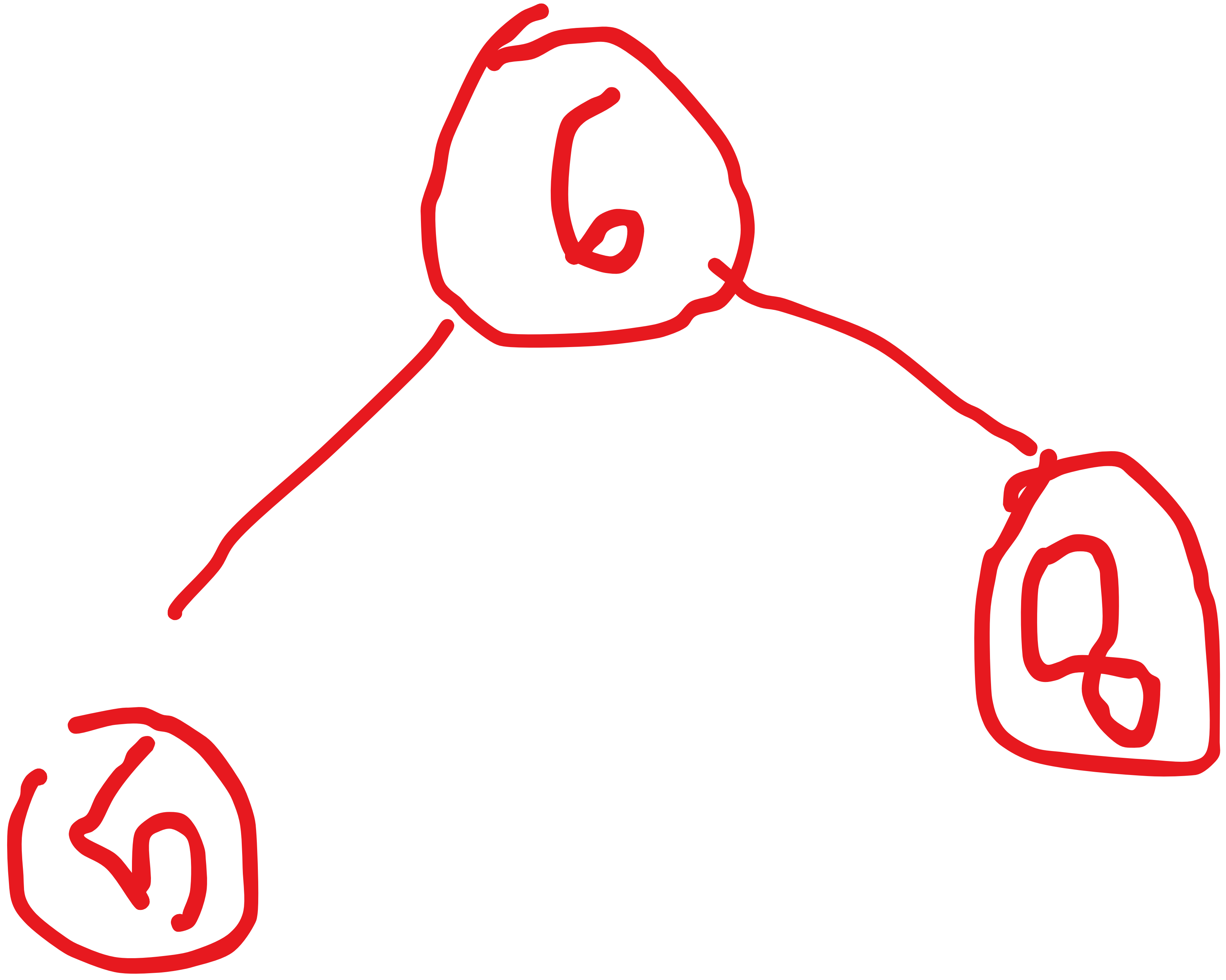

9
5
6



LR/zig-zag

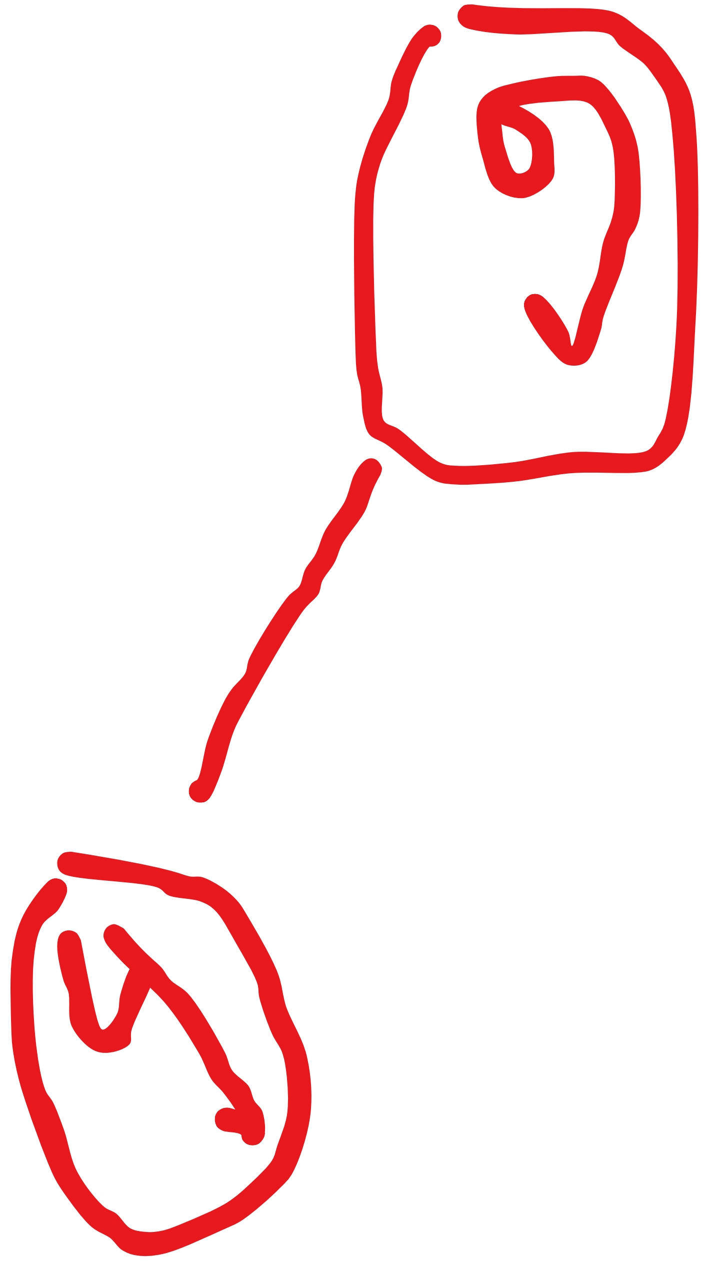

9
5
6
left of parent

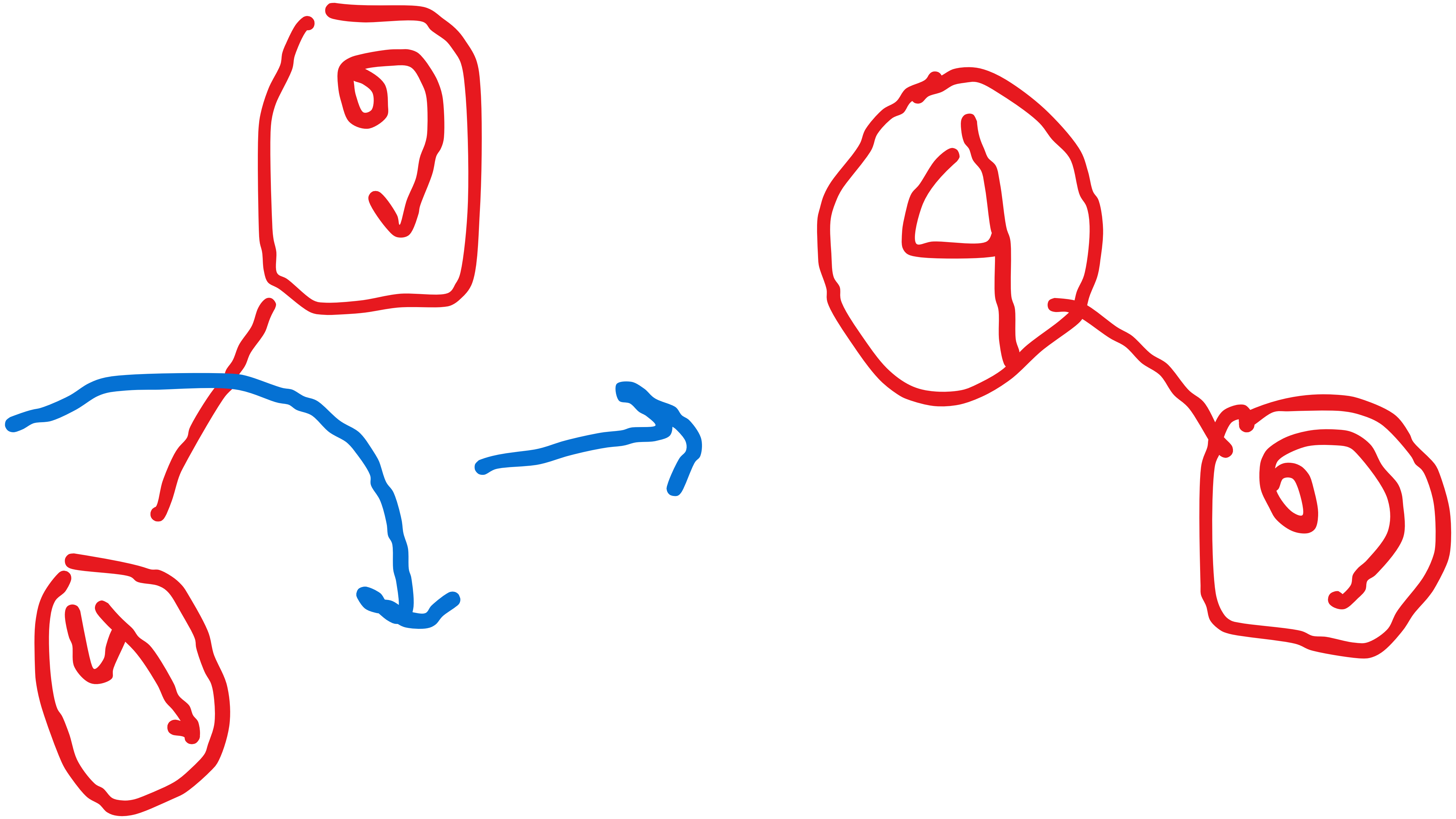

4
6
7
right on parent

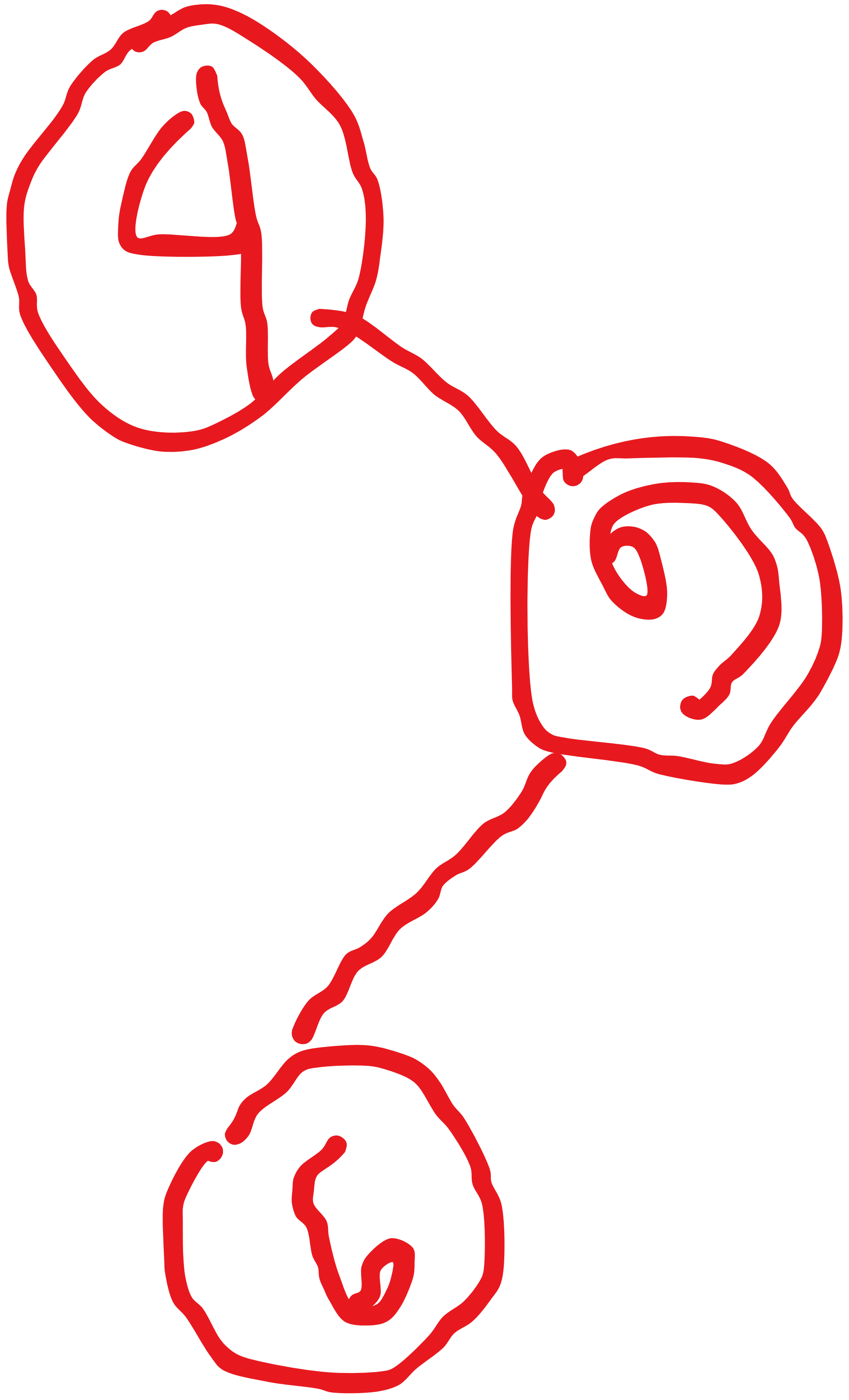

6
5
8

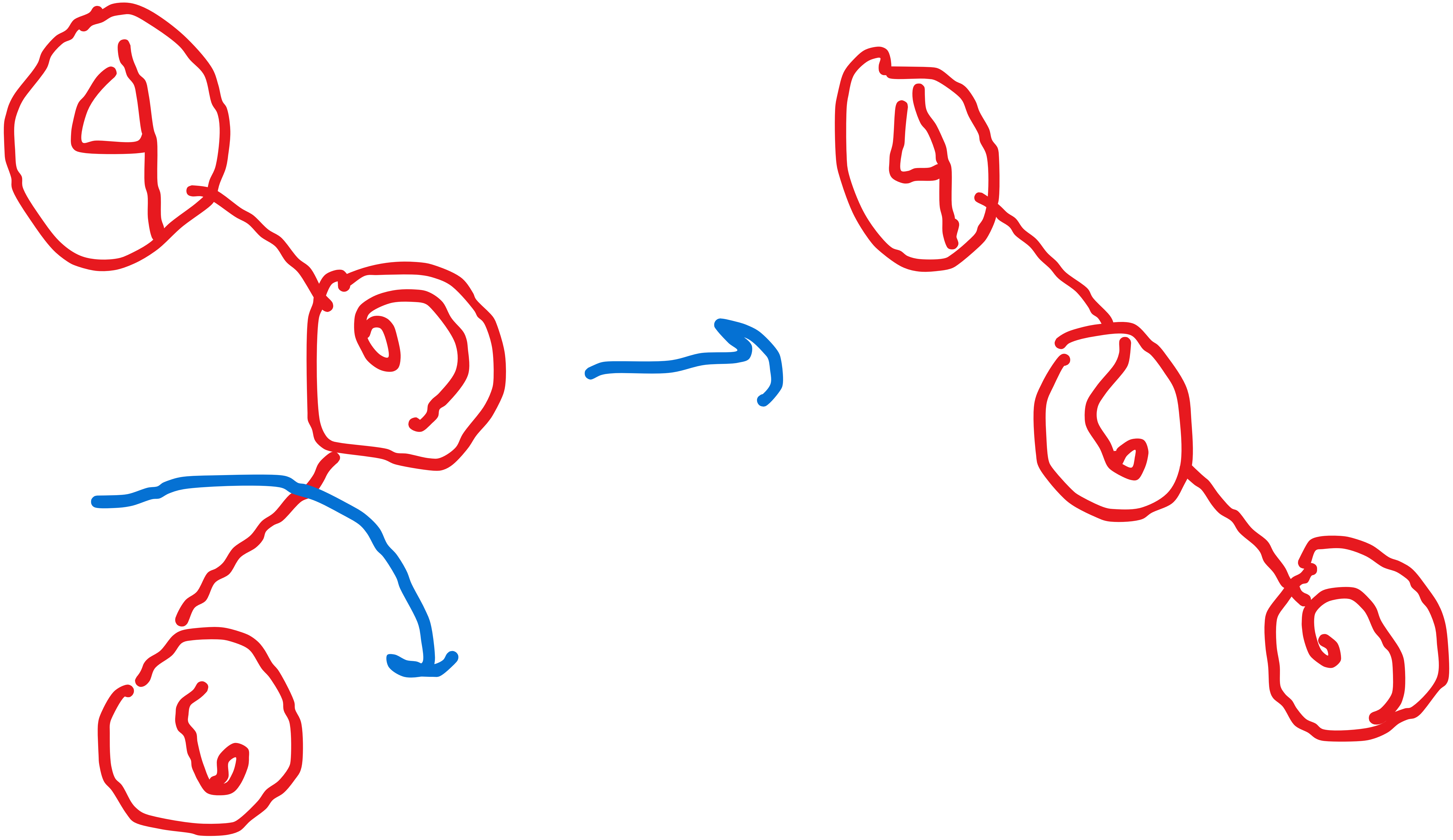
Lets solve an example.

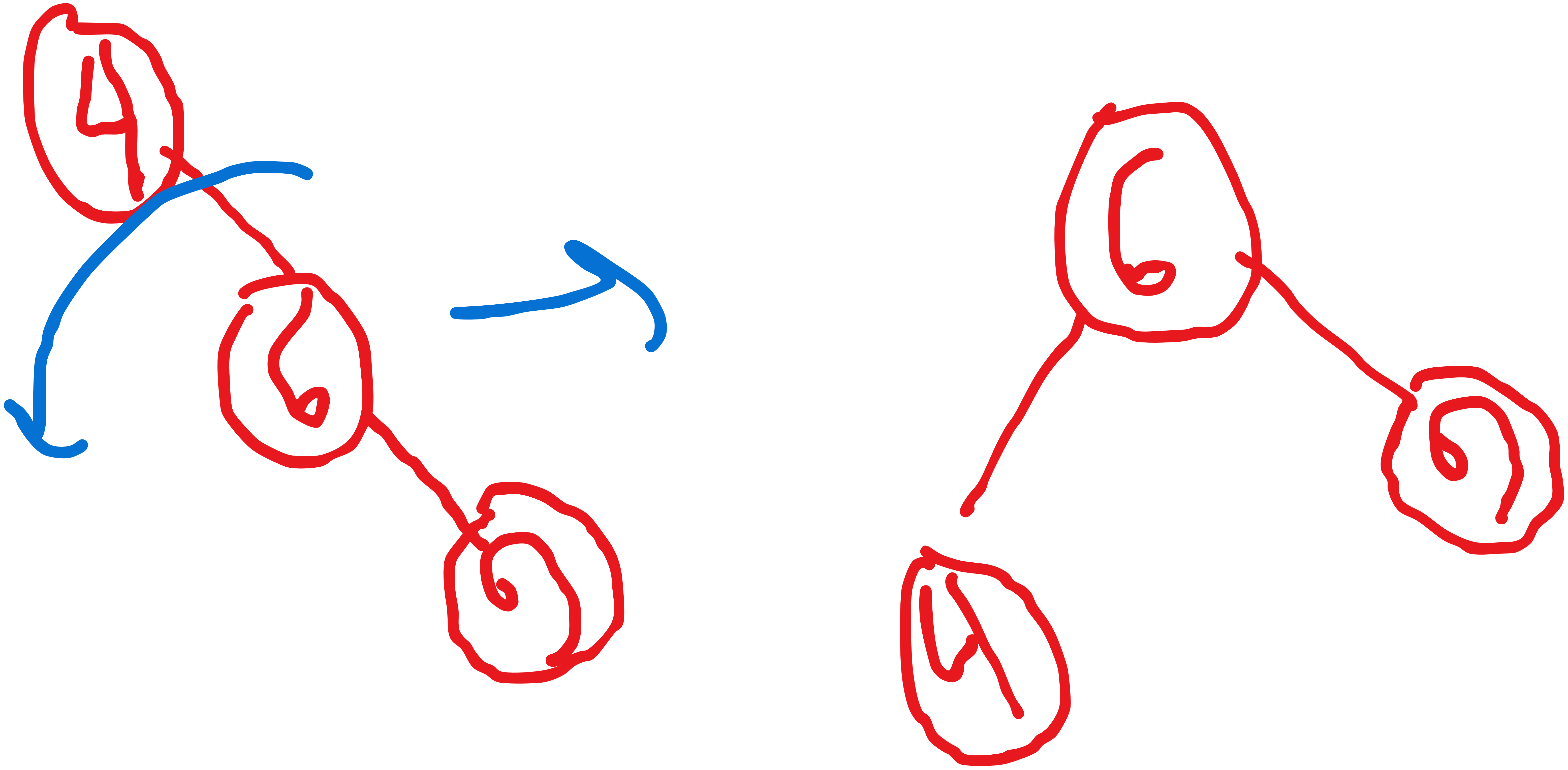
9, 4, 6, 2, 1, 5, 3, 8, 7

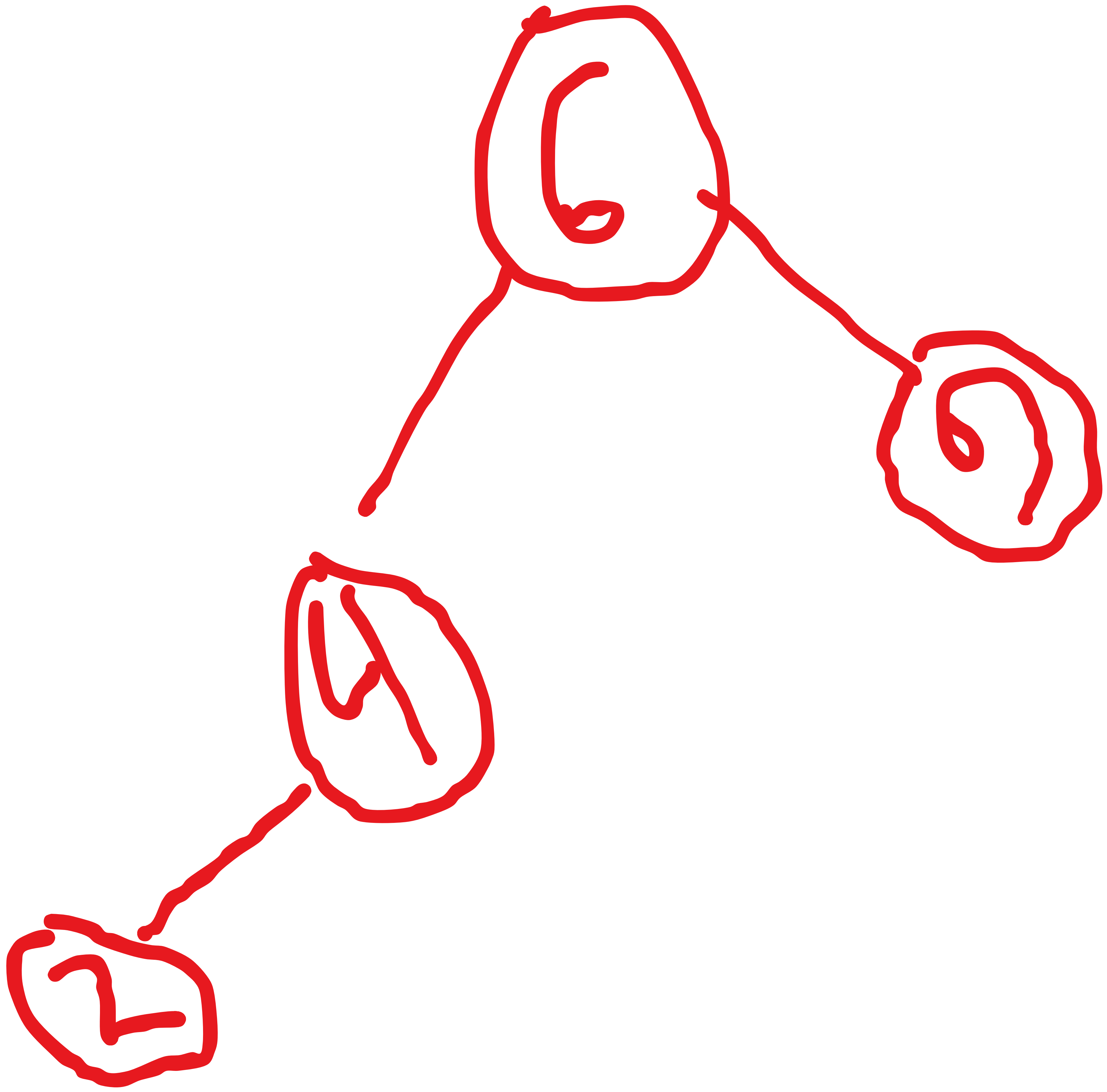

9
4

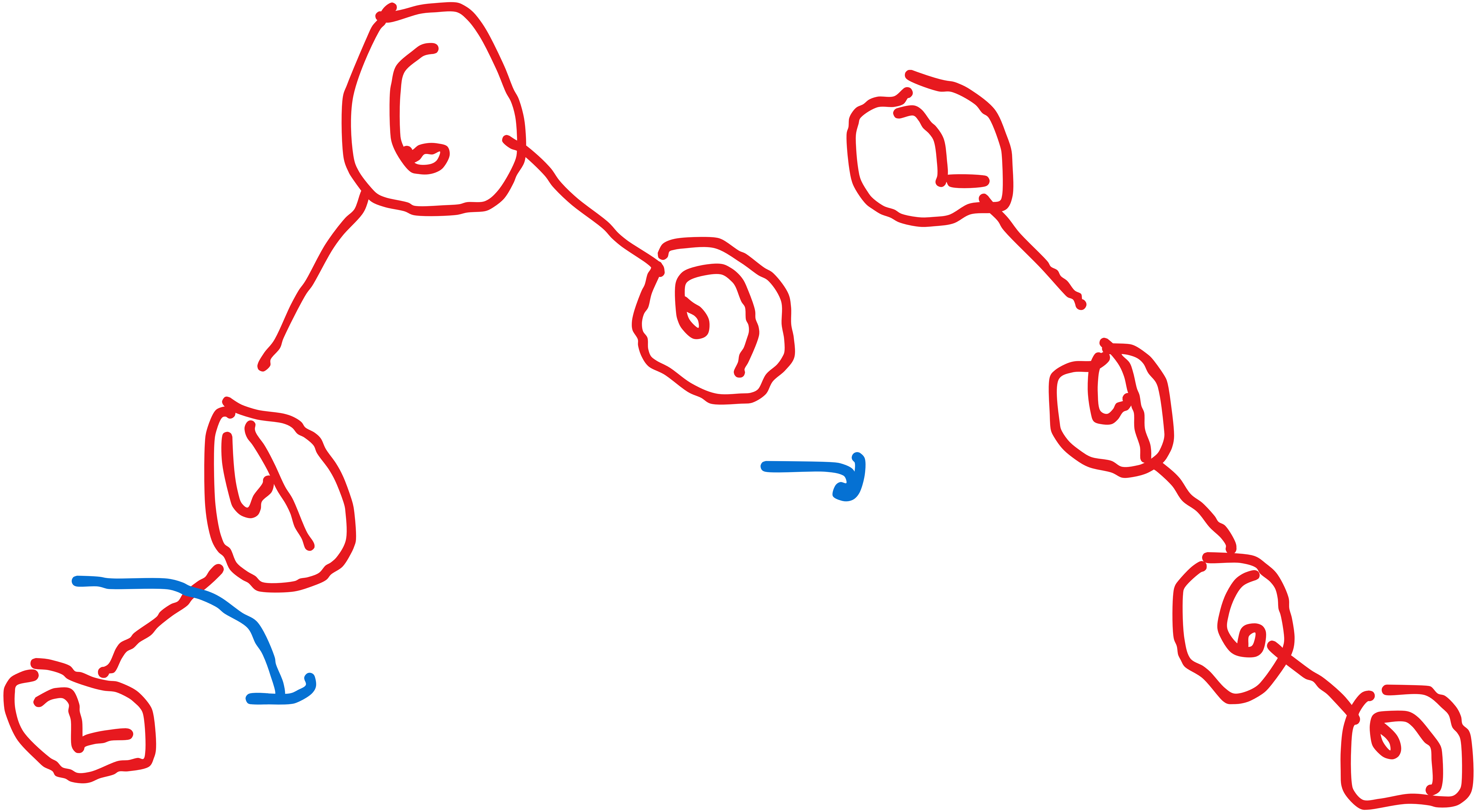

9
4
4
9

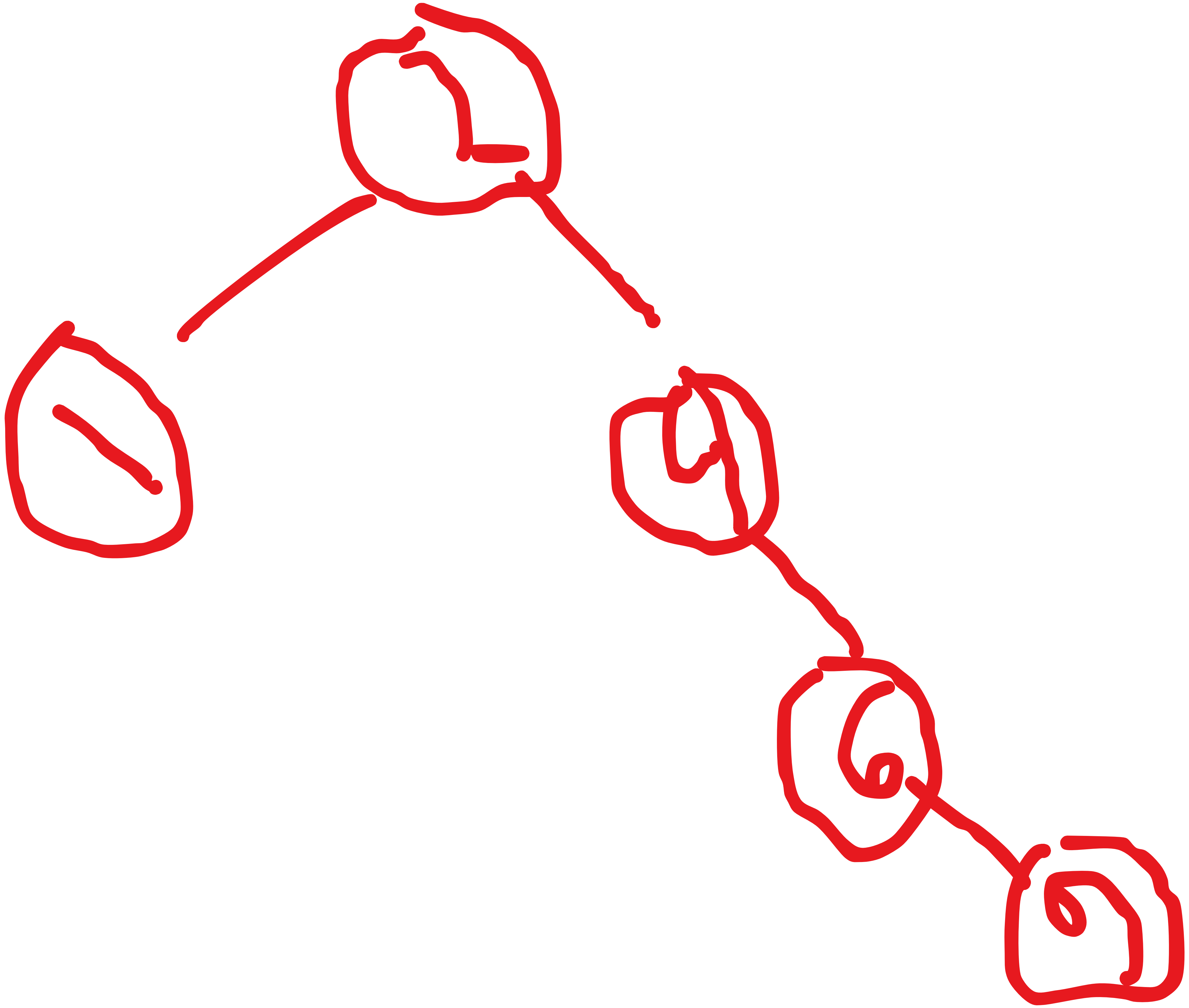

4
6

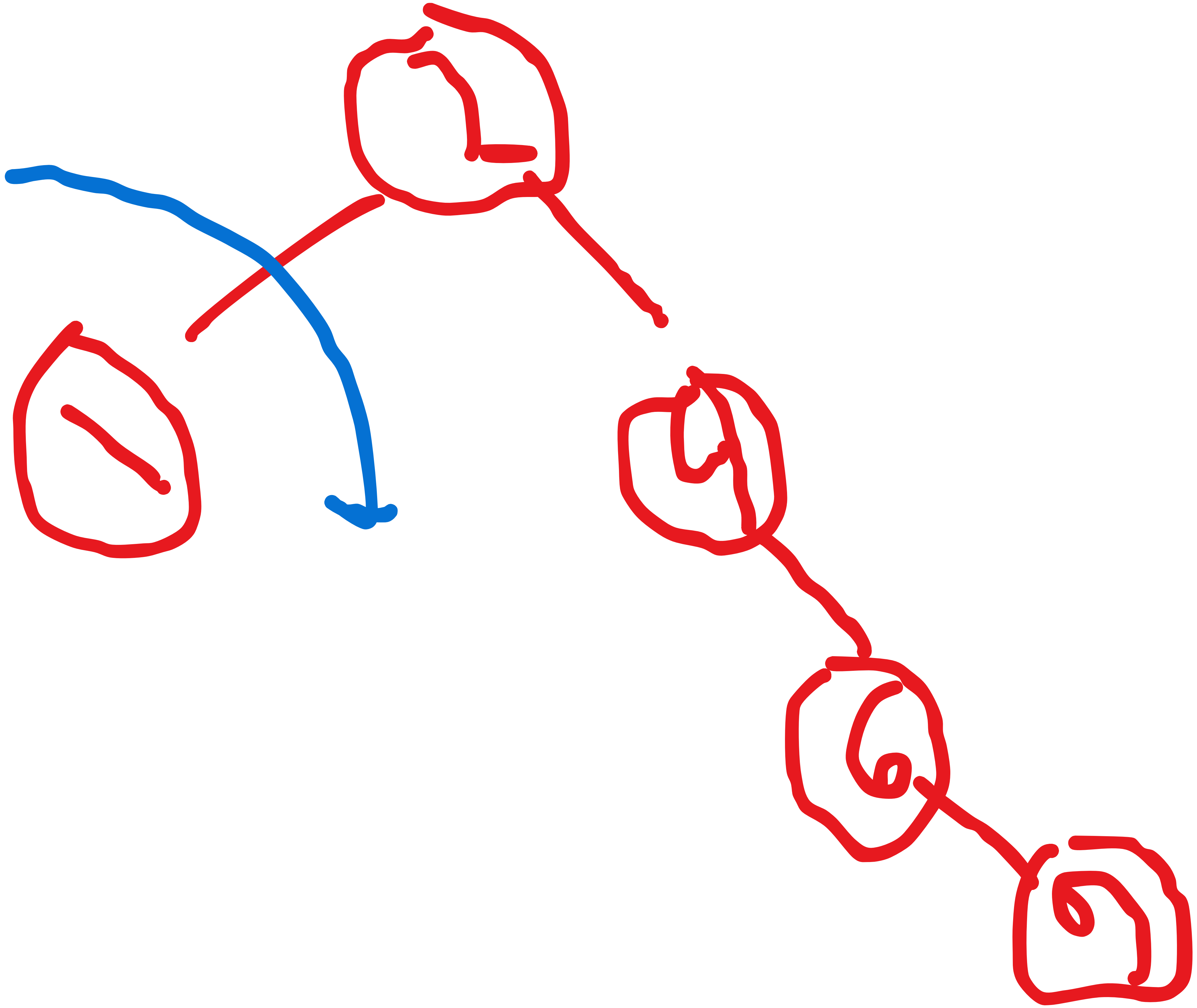

4
9
6
4
6
9

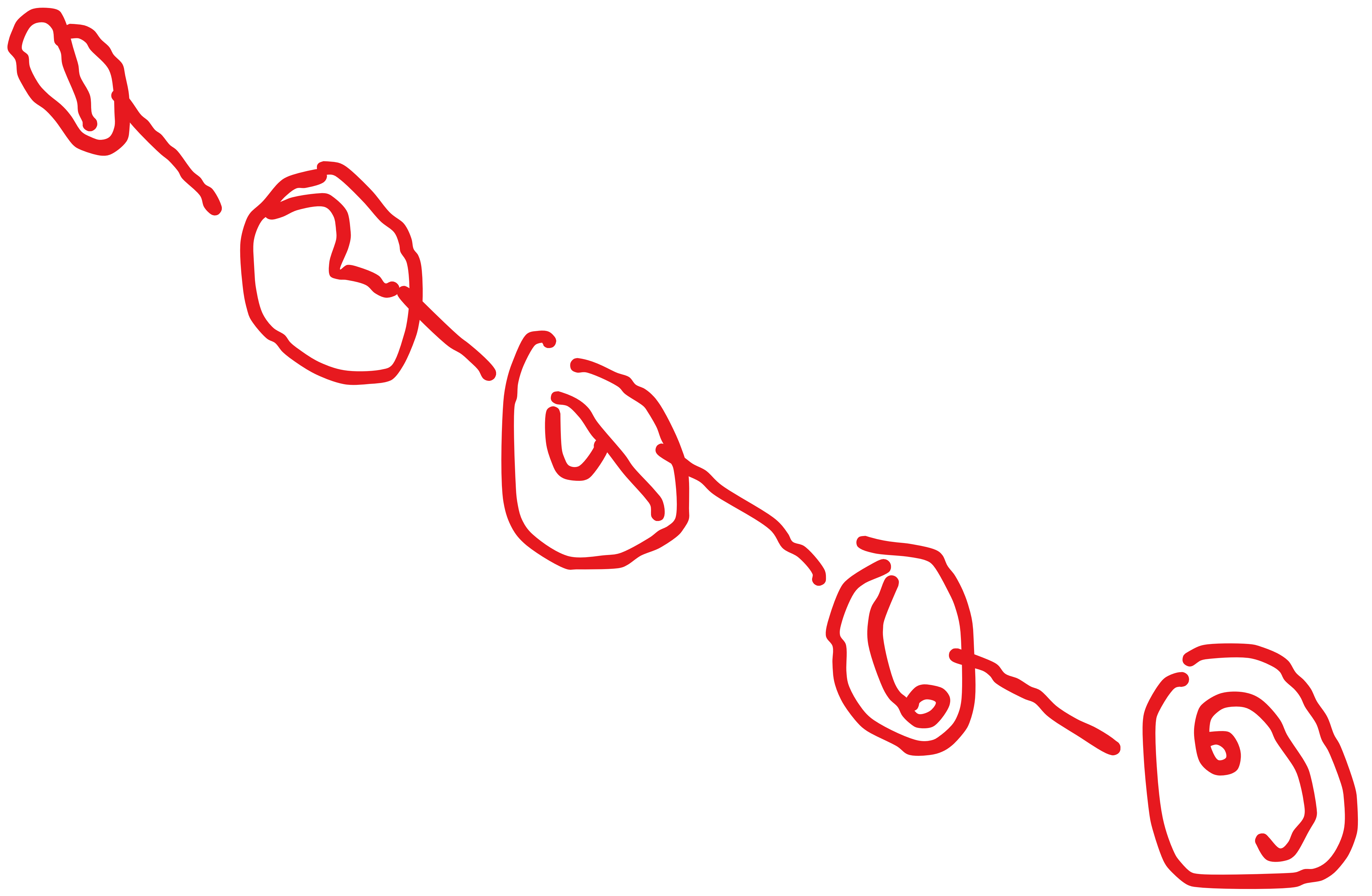

4
6
9
6
4
9

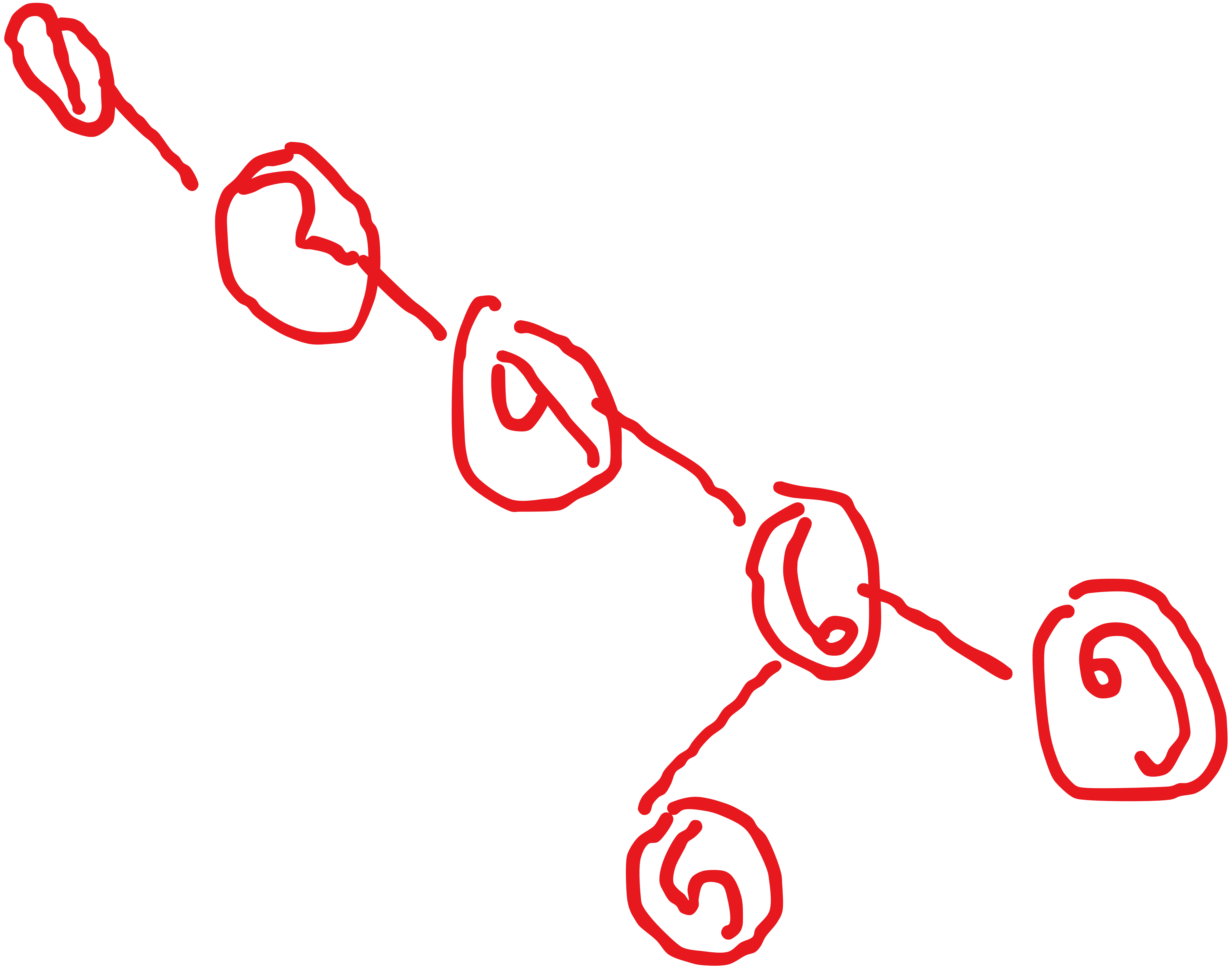

6
4
9
2

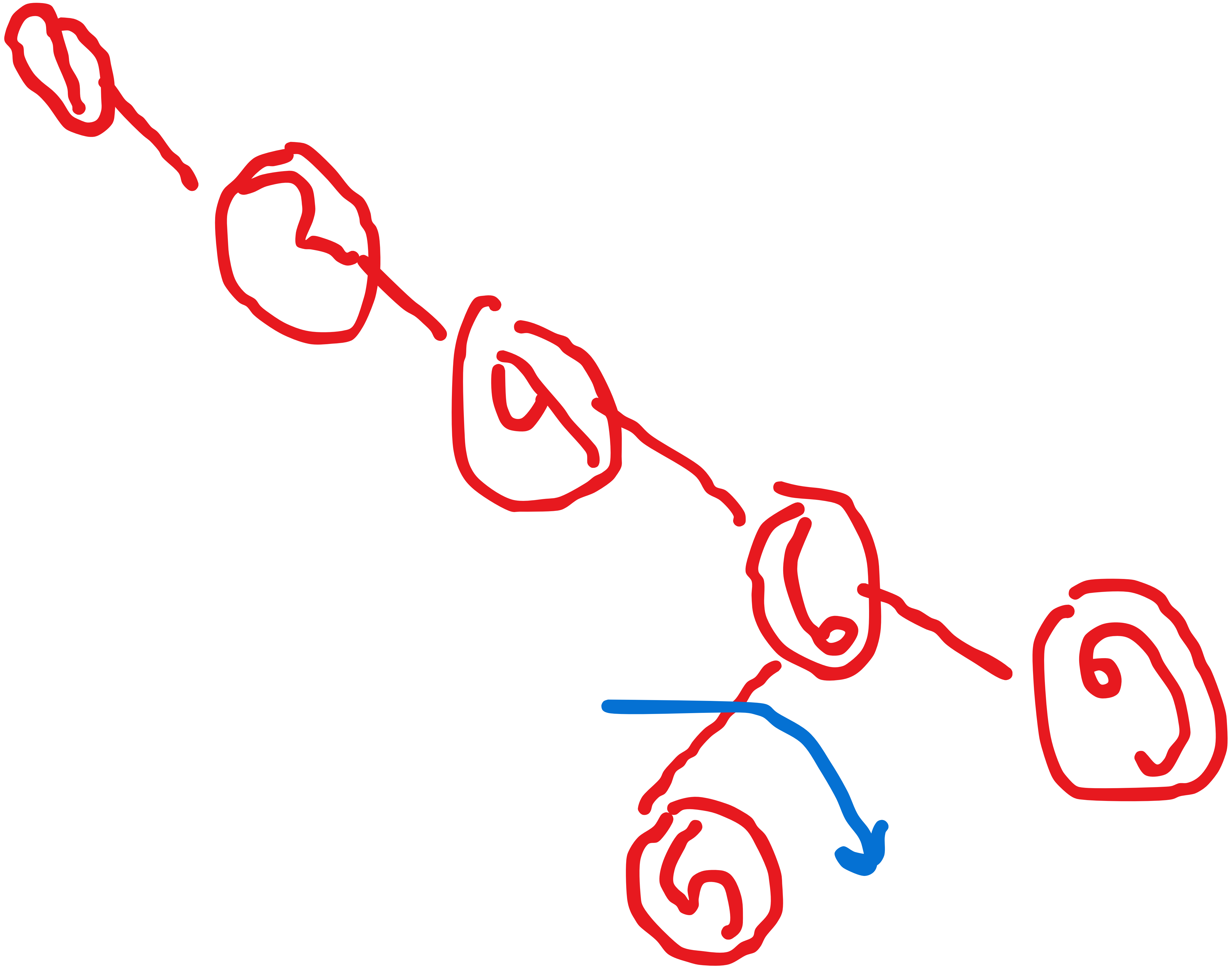

6
4
9
2
2
4
6
9

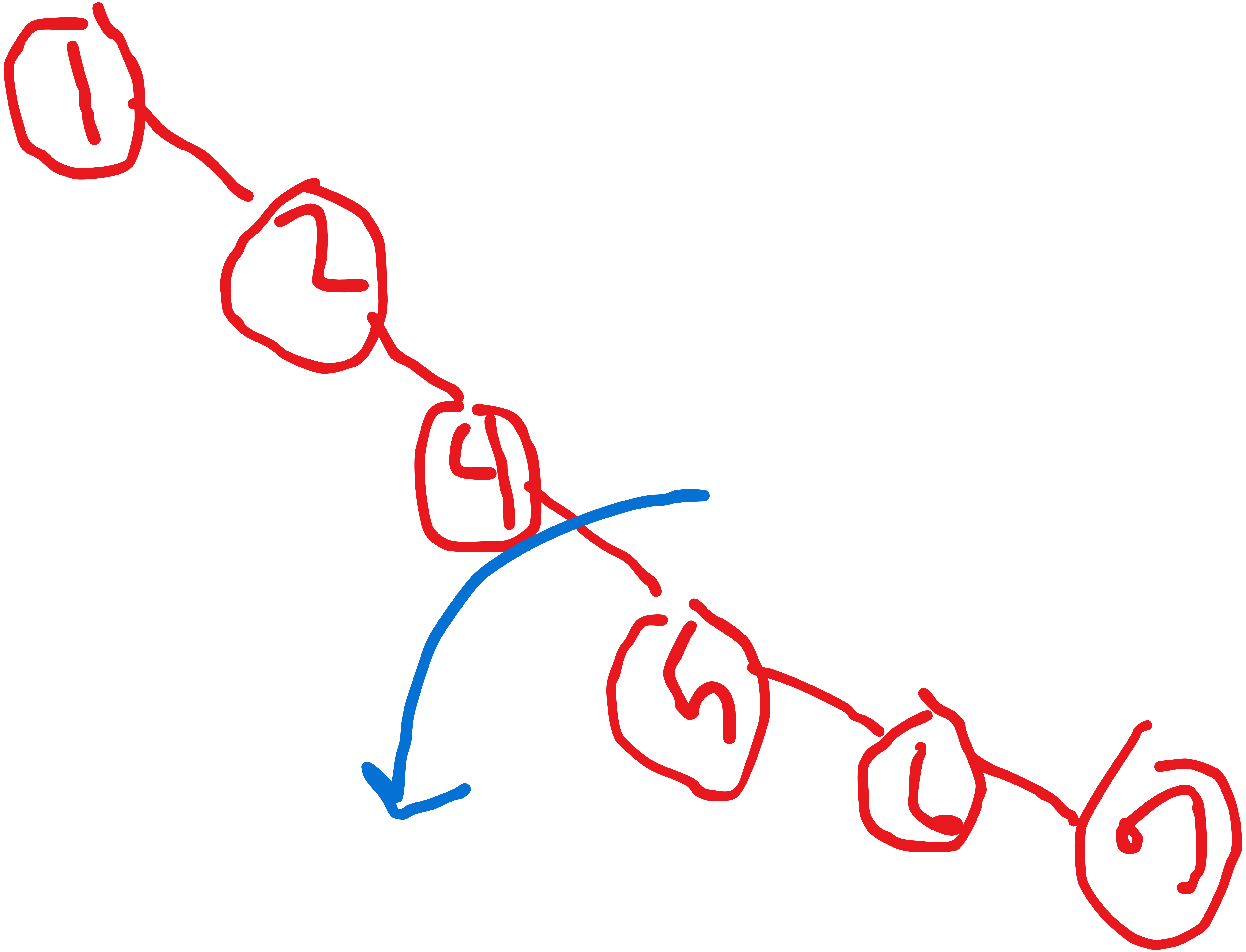

2
1
4
6
9

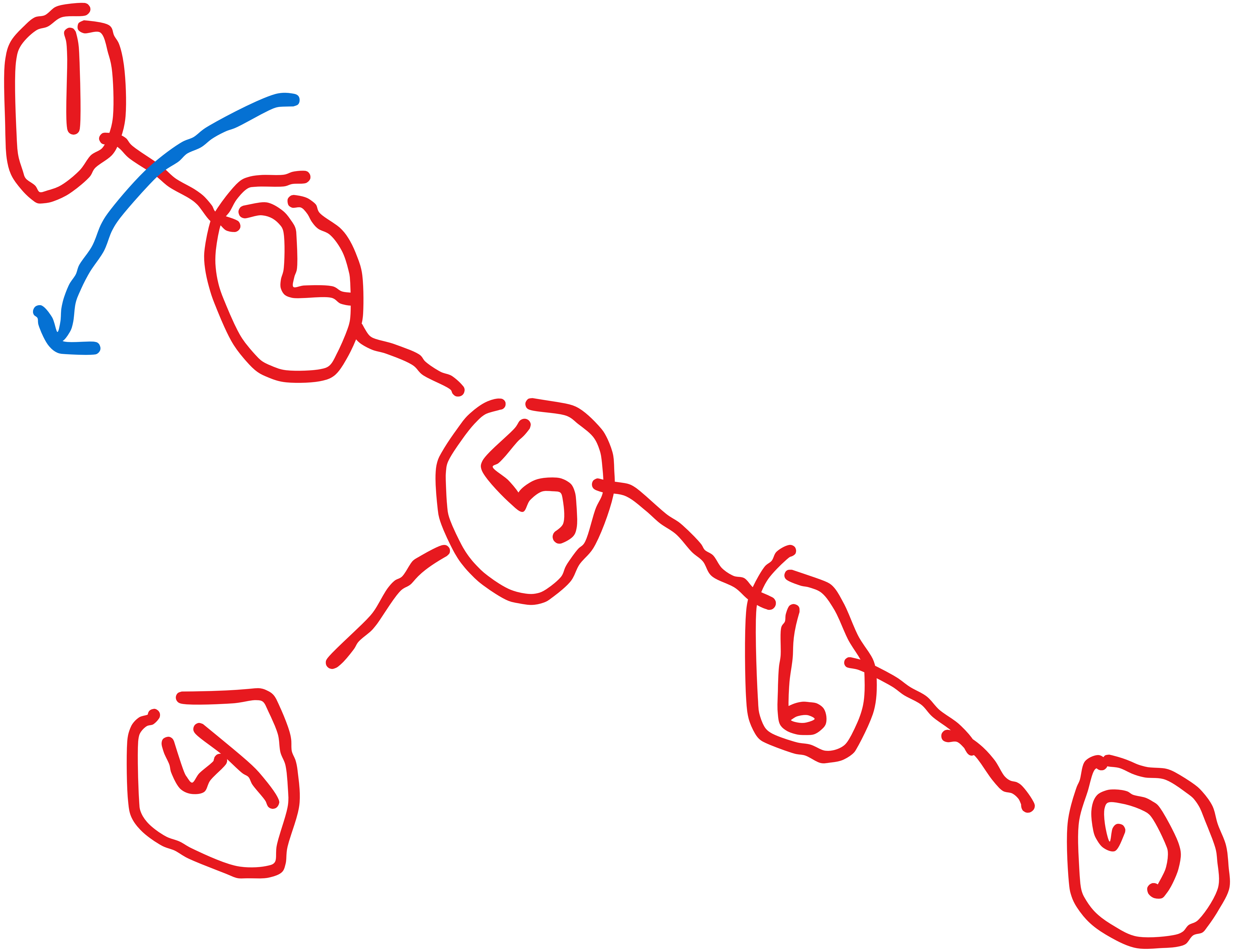

6

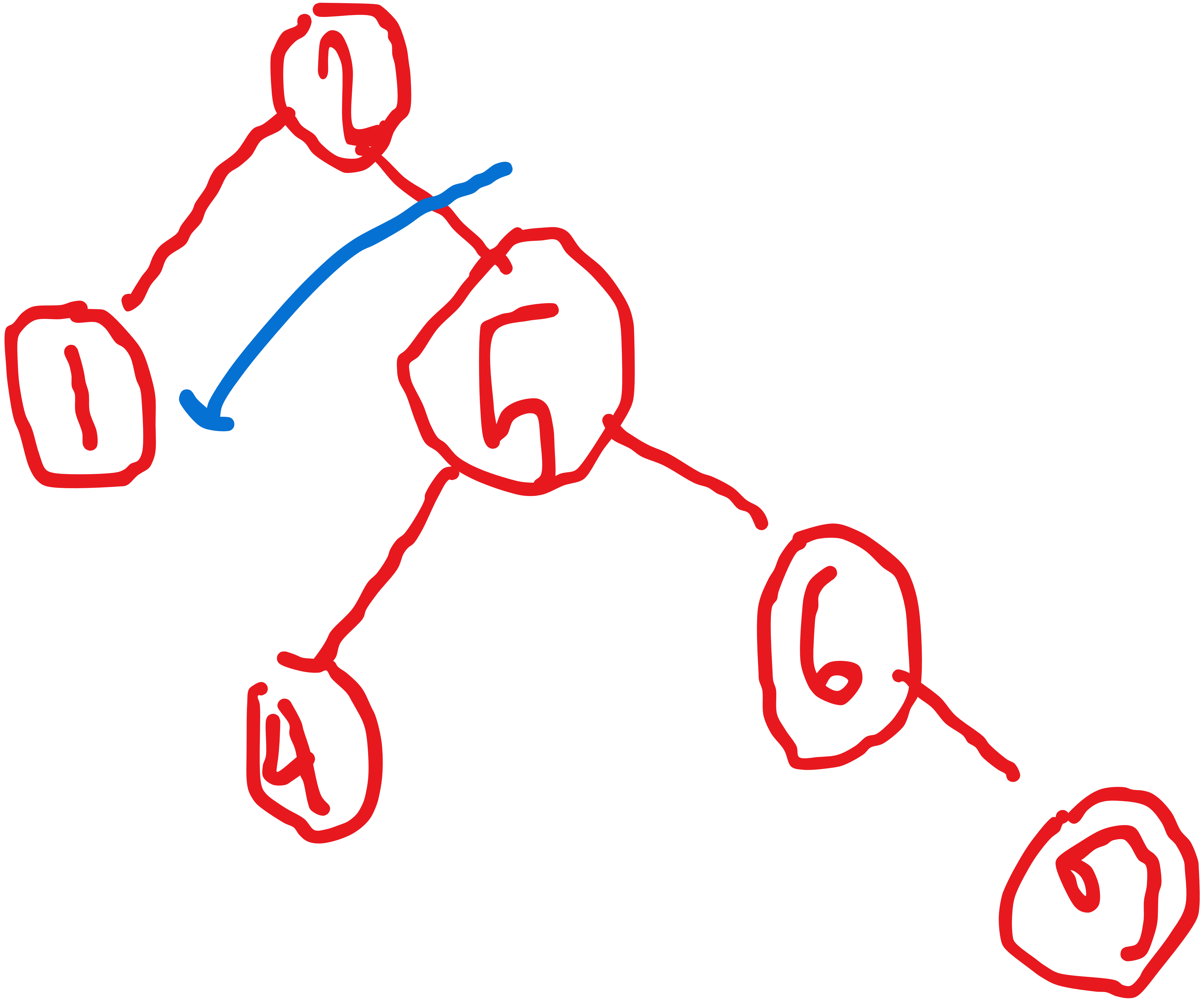

1
2
4
6
9

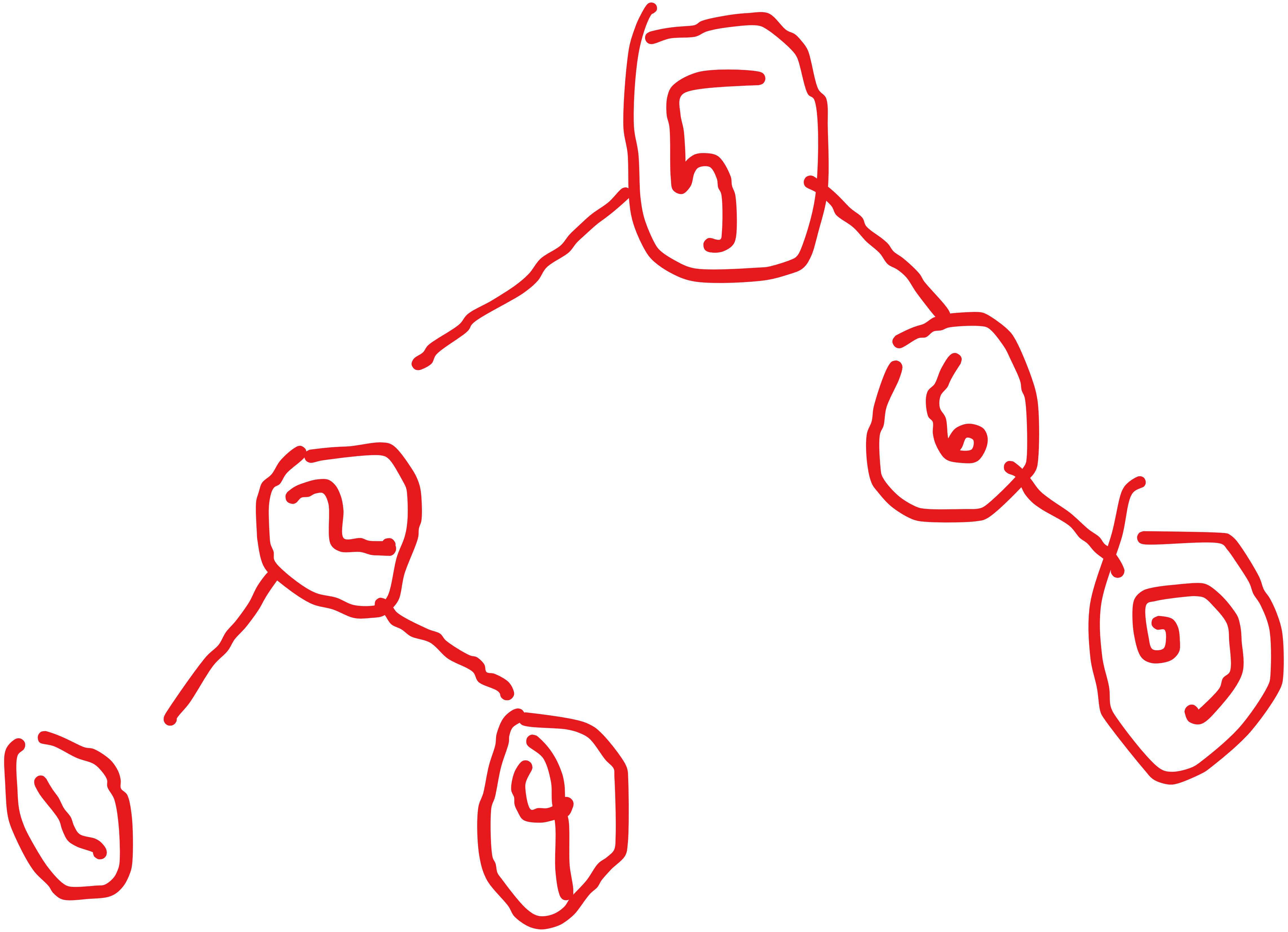

1
2
4
6
5
9

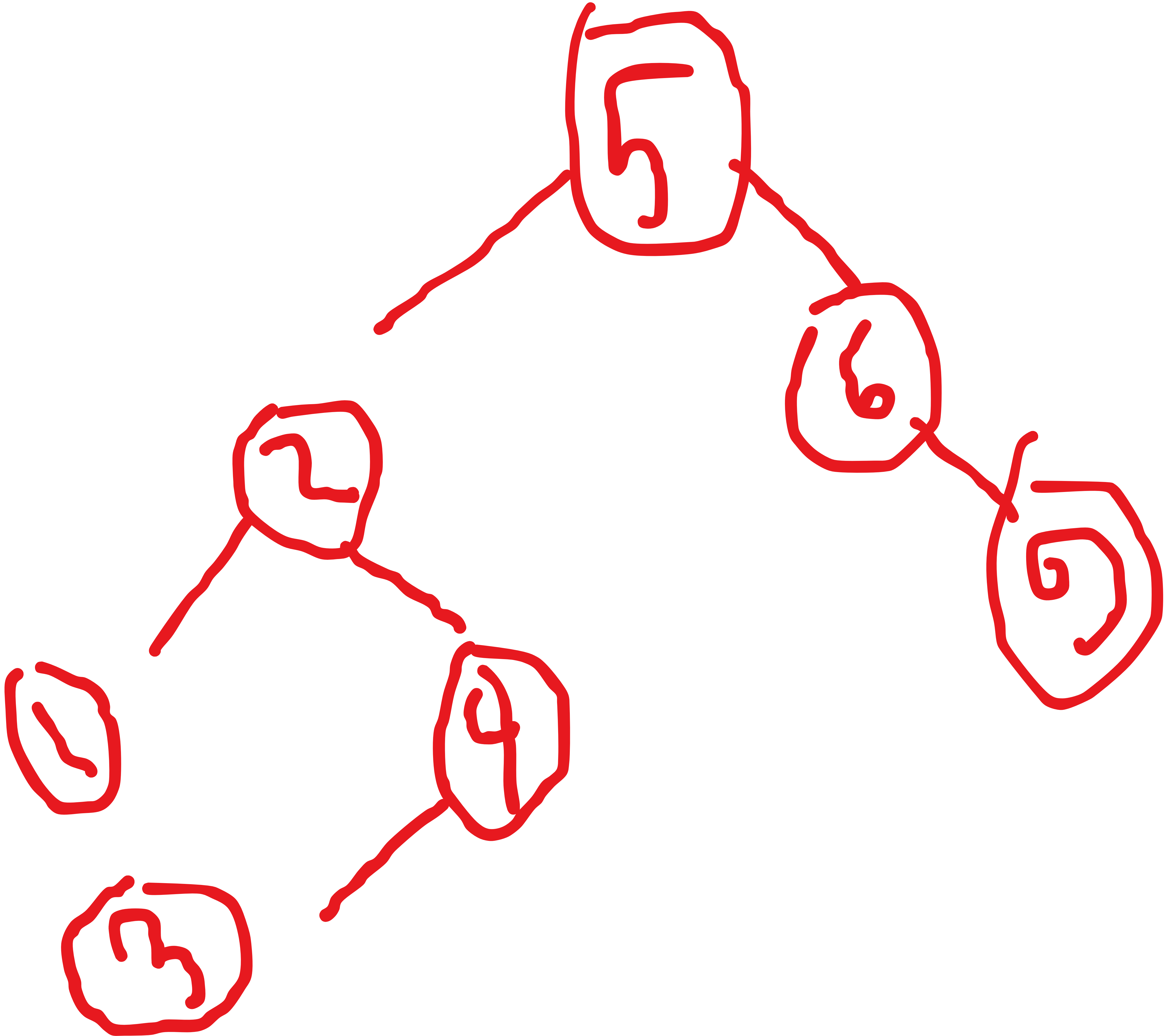

2
4
6
9
5

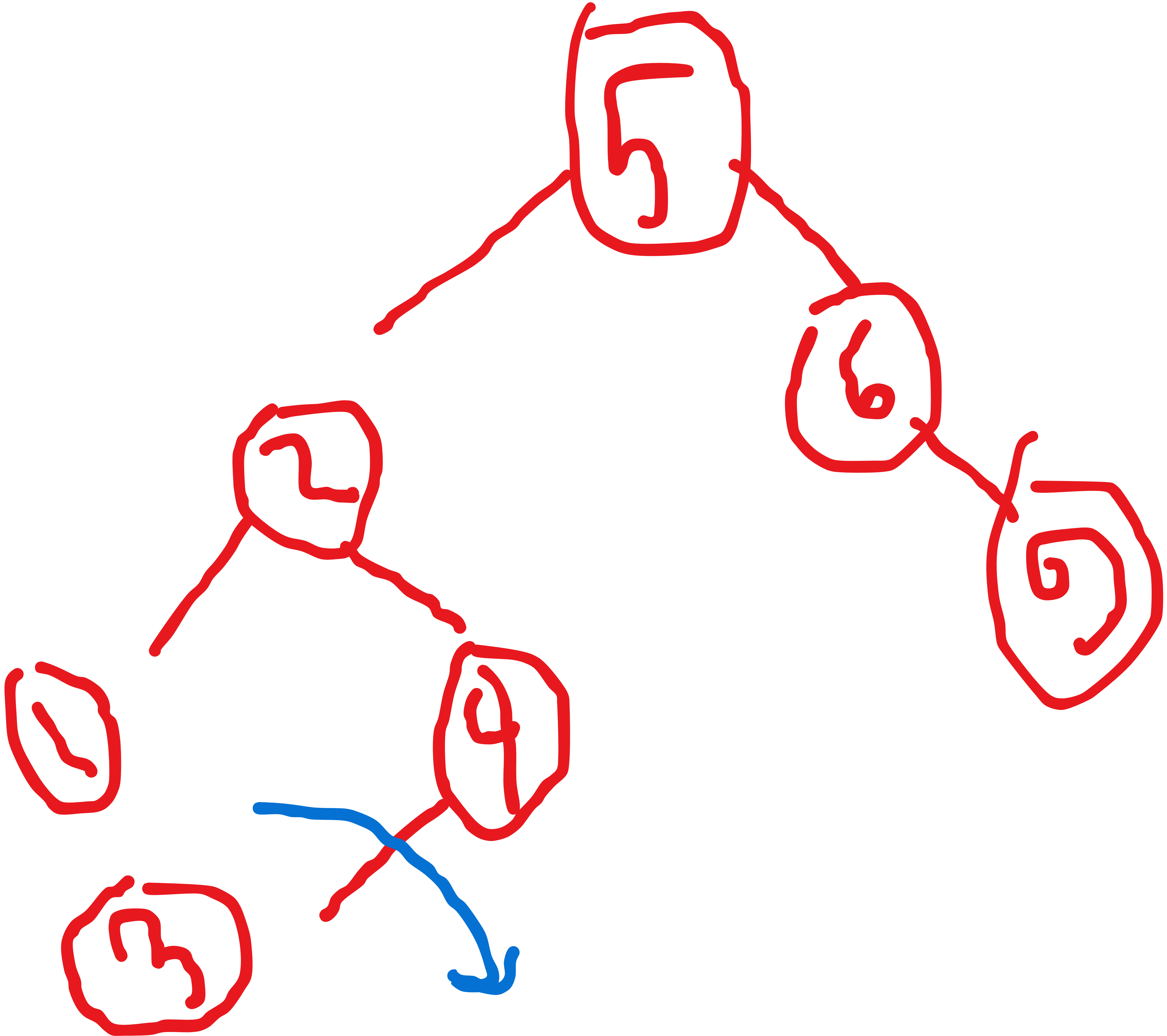

1
2
4
5
6
7


1
2
5
4
6
7

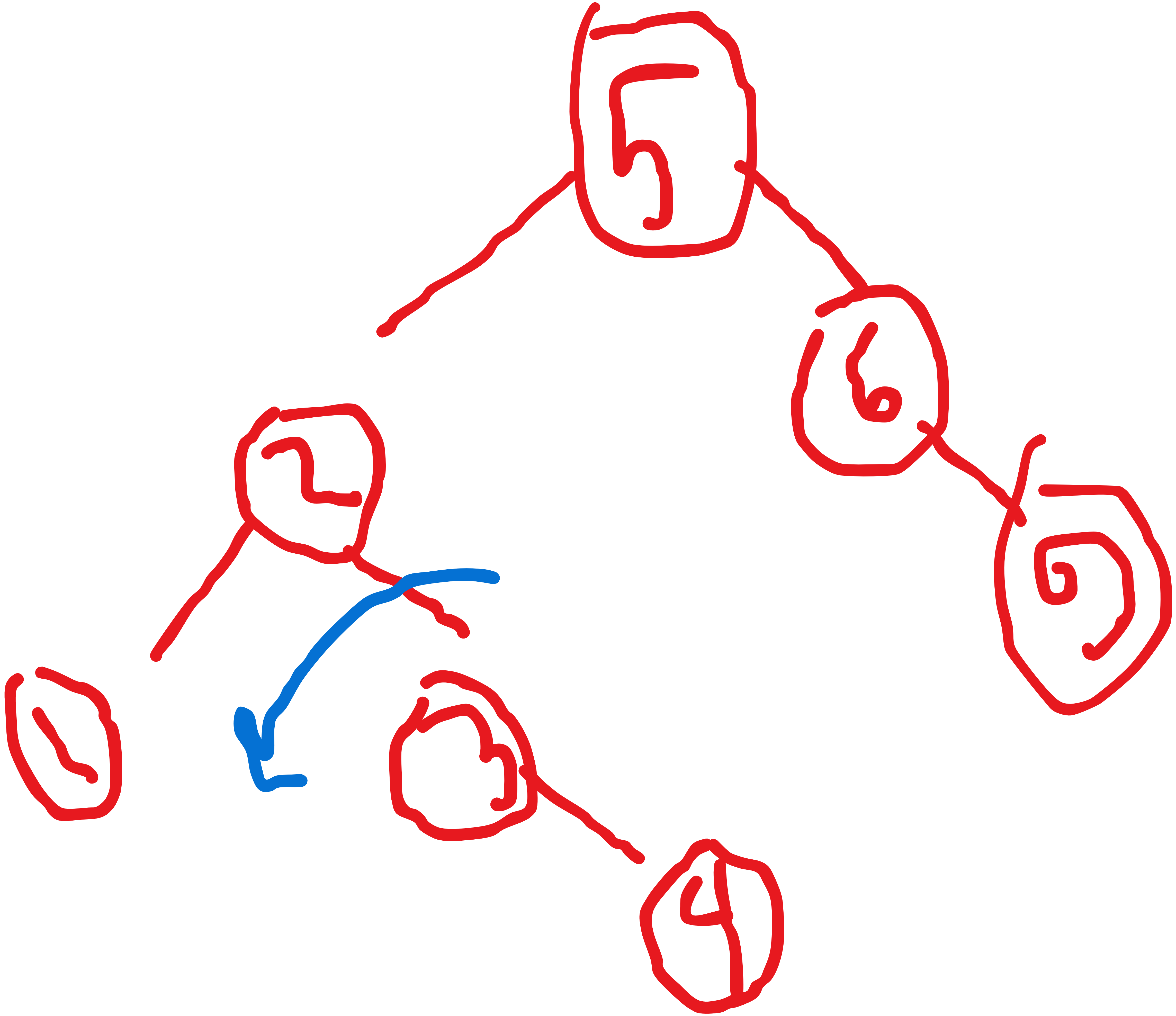

2
1
5
4
6
7

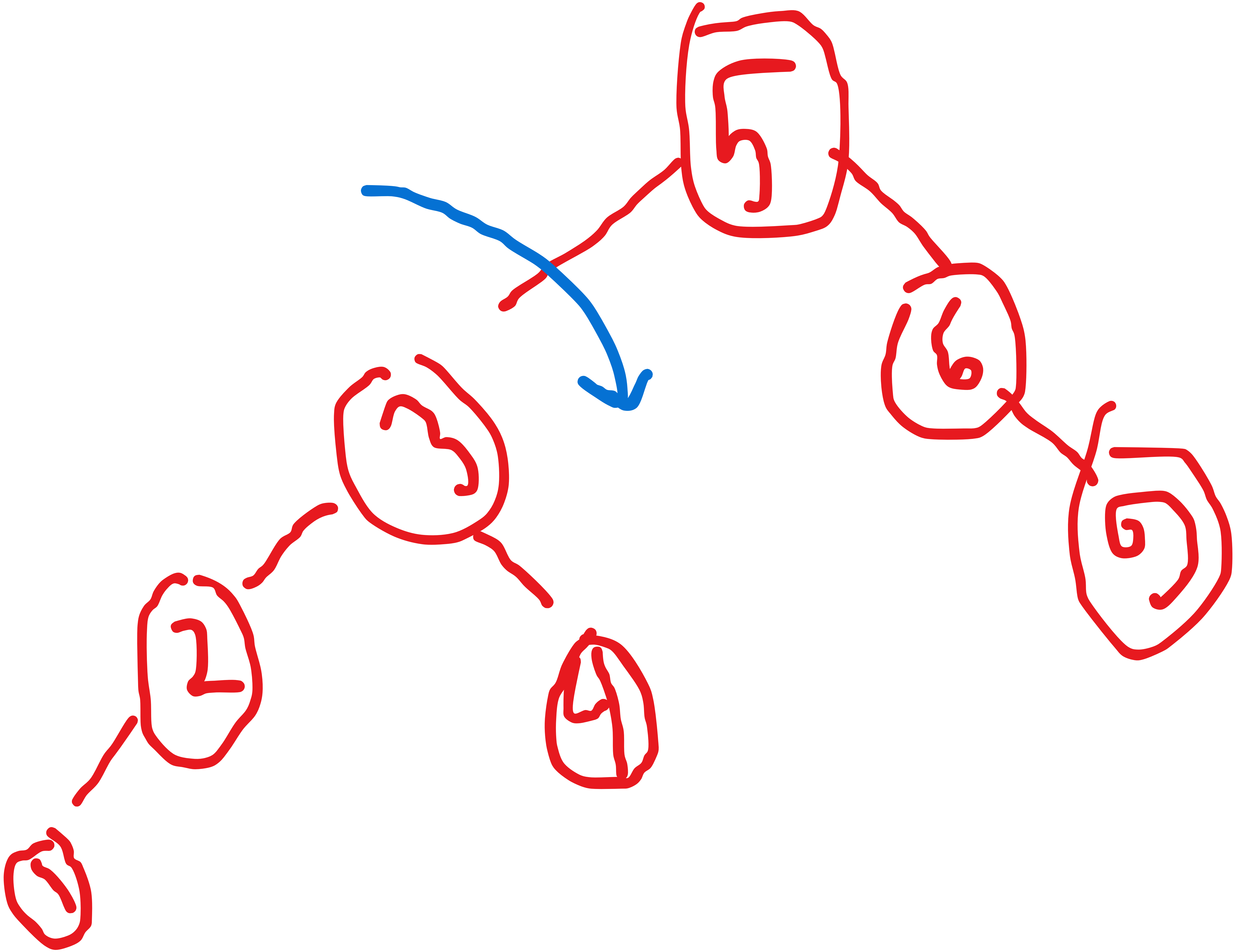

5
2
6
1
4
9

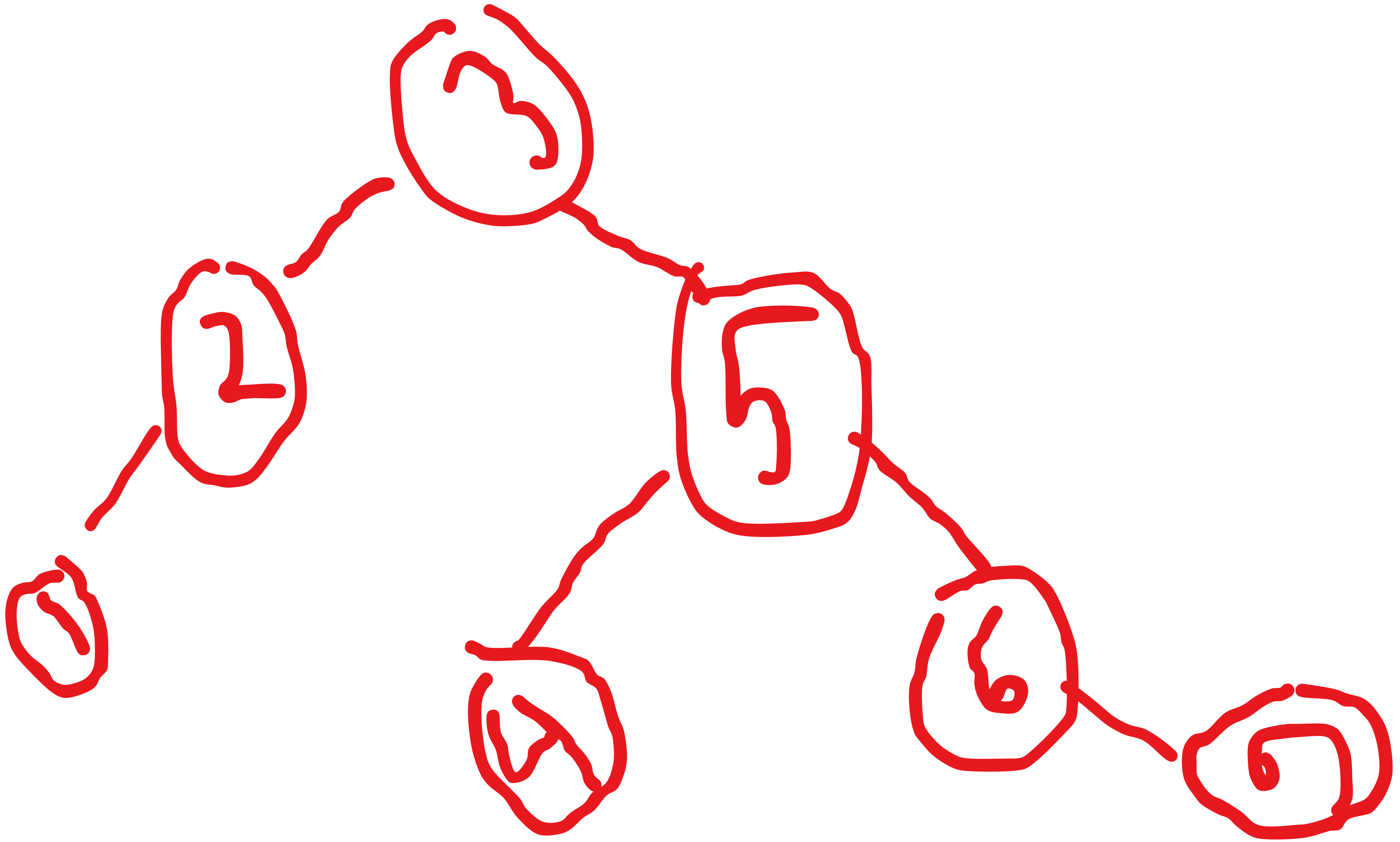

5
6
9
2
1
4
3



5
2
6
9
1
4
3


5
2
6
1
3
4


5
3
6
2
4
9
1


3
2
5
1
4
6
7

Try to add 8 and 7 from your own.

Now go to the implementations