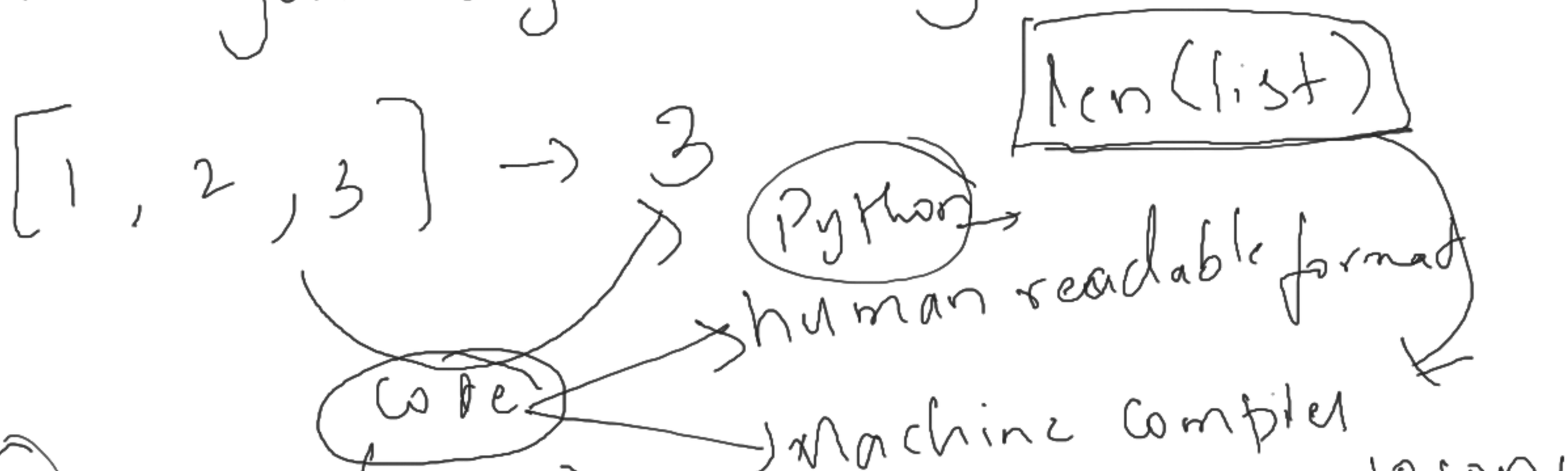
Code your logic → length



1. len(list) = 3

2. list.count(1) = 3 + 0 = 3
   1 : 1, 2 : 1, 3 : 1

(1) len(list) = 3 list = [1, 2, 3]

(2) dict = {x : list.count(x) for x in list}

dict = {1 : 1, 2 : 1, 3 : 1}

values → values → values

sum(dict.values) = 3

(3) own approach.

(Simple)

1. # of lines of code

2. Performance (less time taking) (time Complexity)

D_1 code

100 lines

3 sec

D_2

80 lines

6 sec

2nd list = [1, 2, 3]

start = time()

len(list)

end = time()

diff = end - start

print(diff) = 0.03 sec

(1)

(2)

sum

2nd ~~dict~~ logic

start = time()

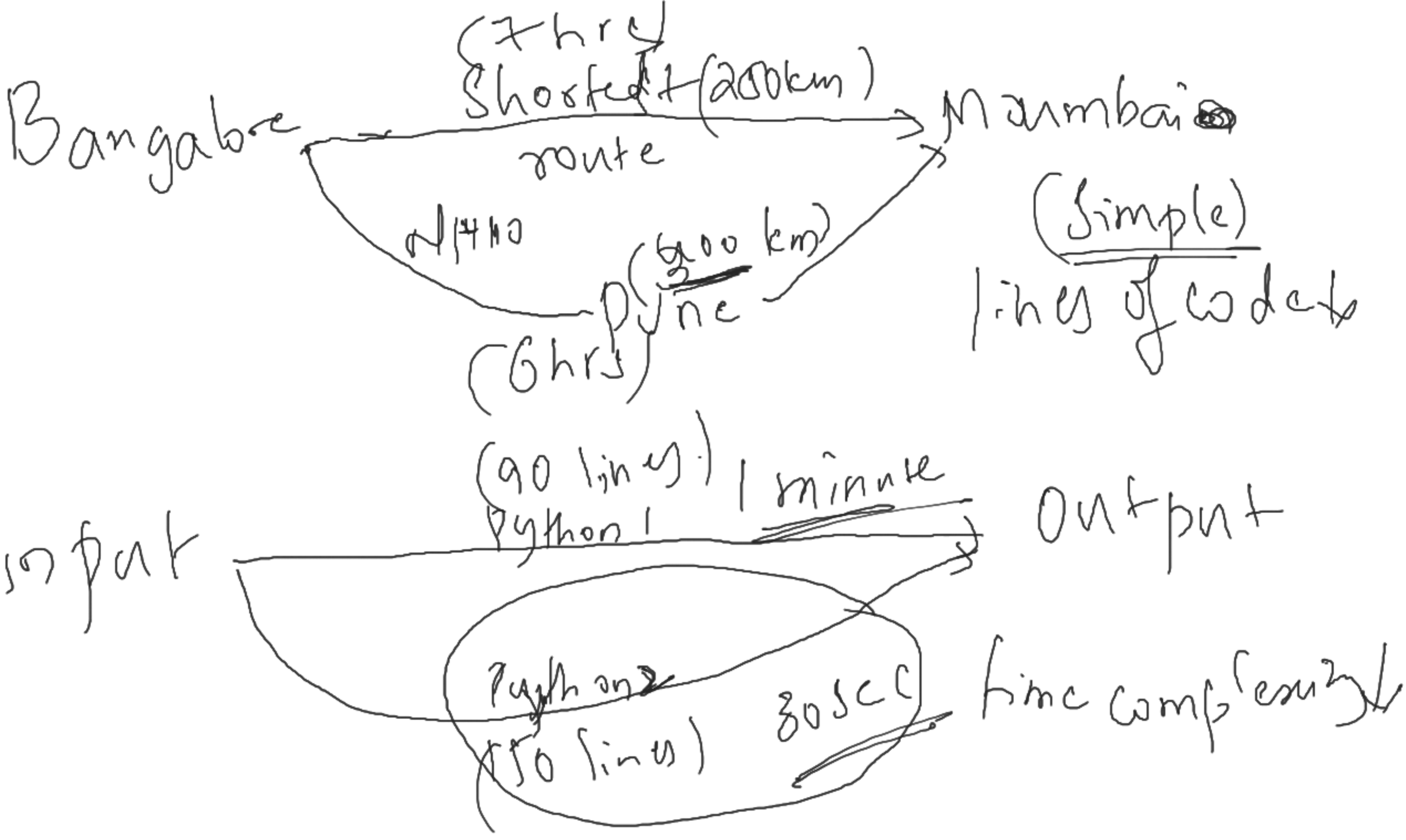
count

(dict values)

end = time()

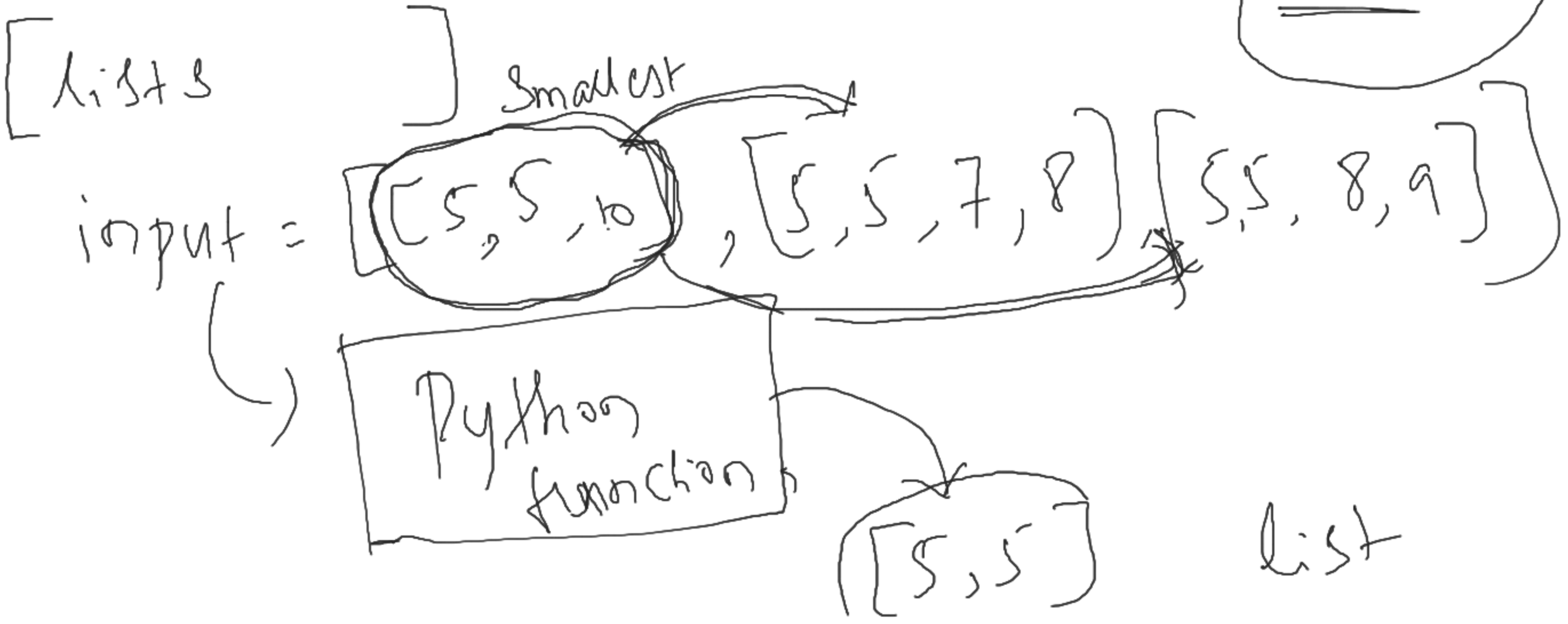
diff = end - start

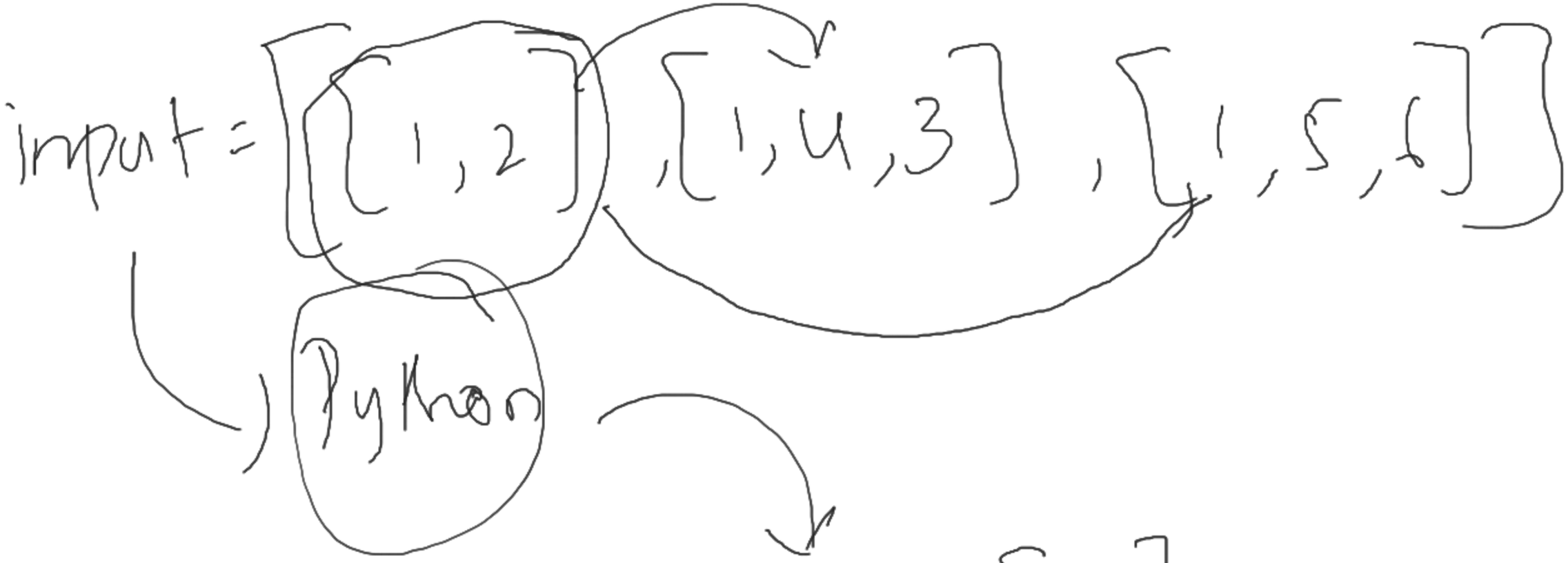
diff = 1 sec


(7hrs)
Shortest (250km)
route
Bangalore
Mumbai
NH10
(600 km)
Pune
(6hrs)
(Simple)
lines of code
(90 lines)
Python
1 minute
input
Output
Python
(50 lines)
Bosch
fine comprehension

1. time
2. Simple (less lines of code)

[ ], [ ], [ ] -> list of lists -> nested list





input = [[5,6,8], [1,5,8,6,2], [5,2,6,8]]

output = [5,6,8] = [8,6,5] = [6,8,5]

input = [[1,2],[1,2,3],[4,5]]

Output = [ ] = []

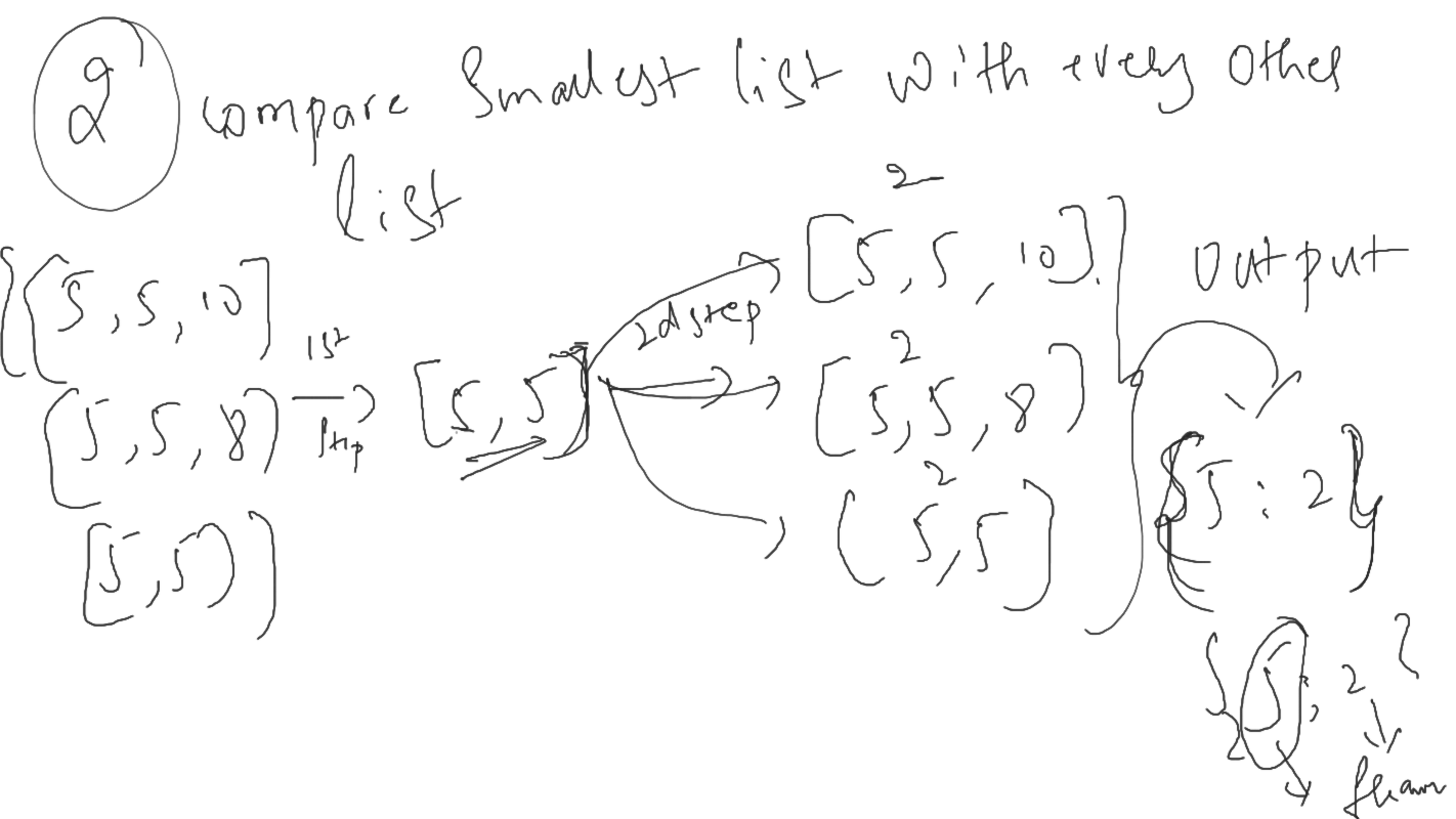
= [5,5]

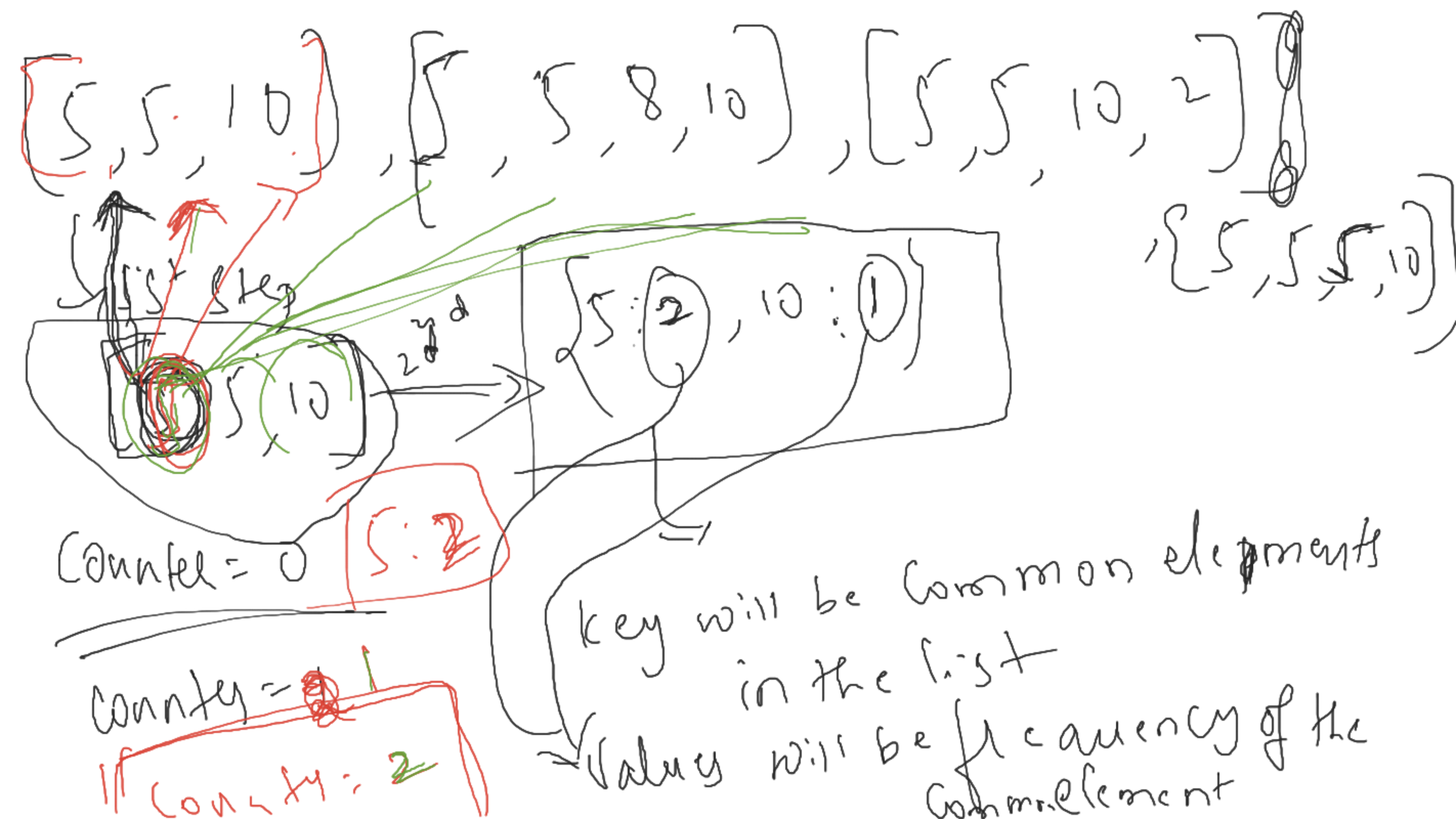
① find the Smallest list.

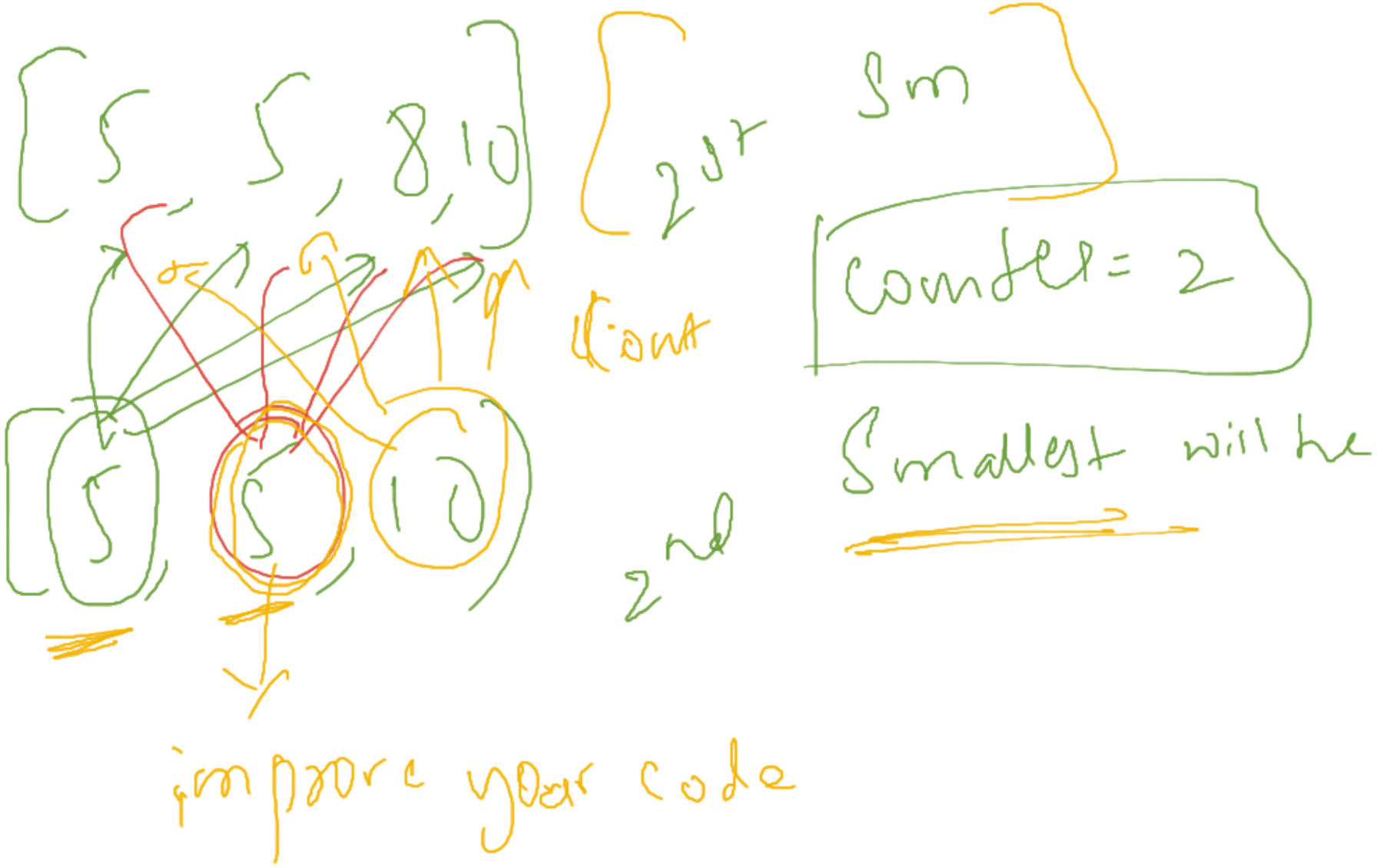
input = [[5, 5, 10], [5, 5, 8], [5, 5]]

→ function (logic) → output

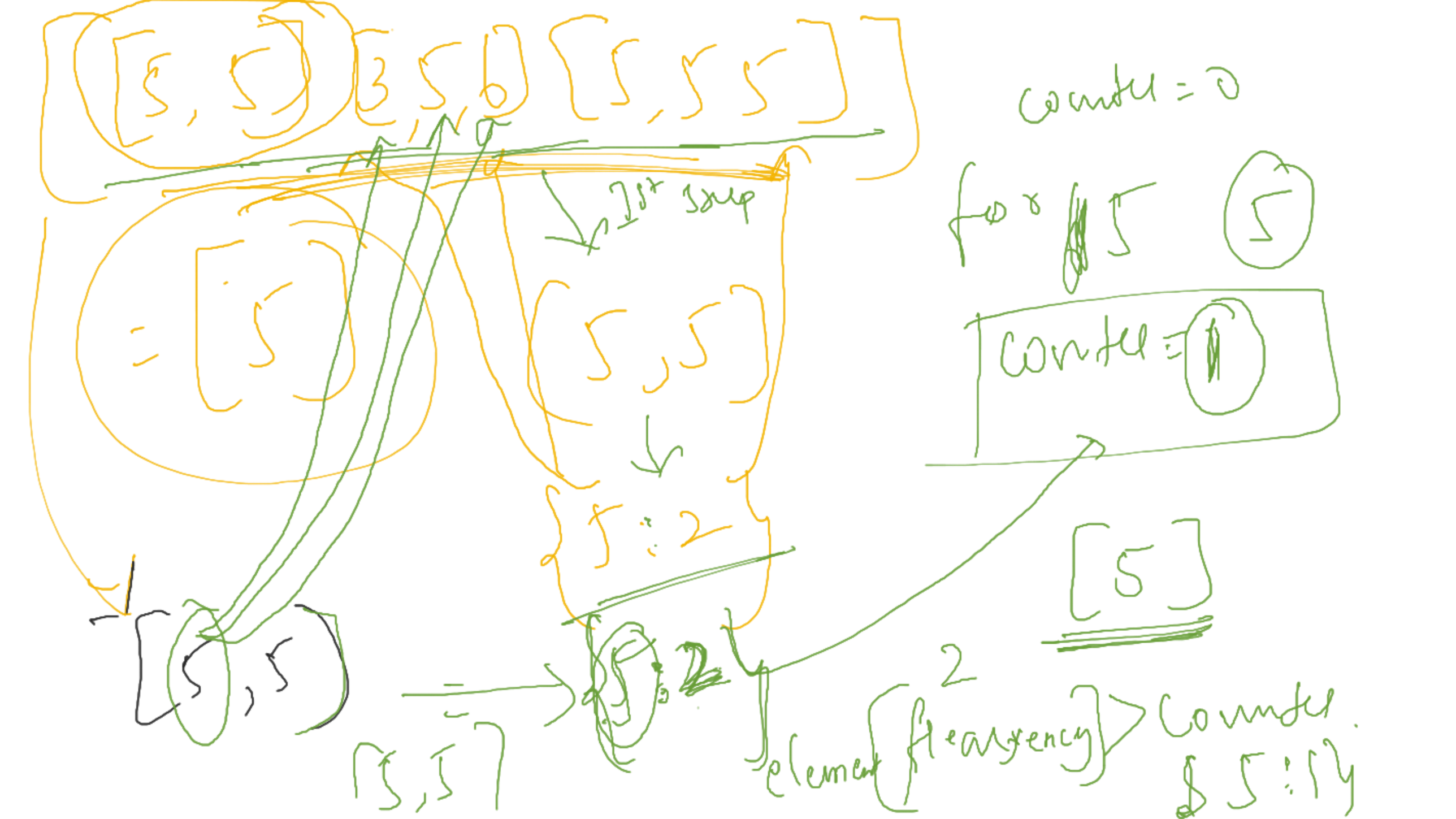
[5, 5]

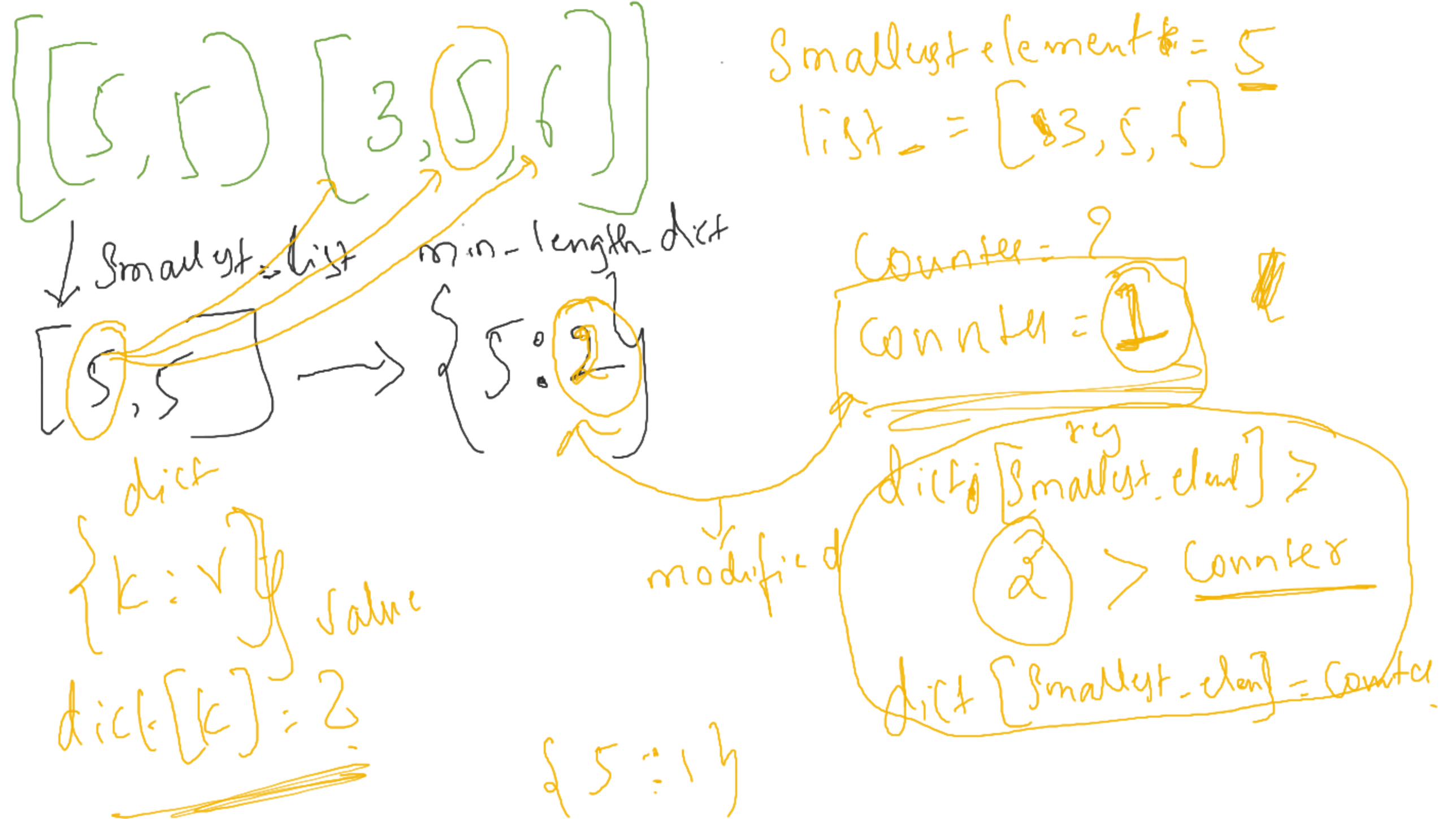

2
compare Smallest list with every Other list
[[5,5,10]
[5,5,8]
[5,5]]
1st step
[5,5]
2d step
[5,5,10]
[5,5,8]
[5,5]
Output
{5:2}
{5:2}
flaw

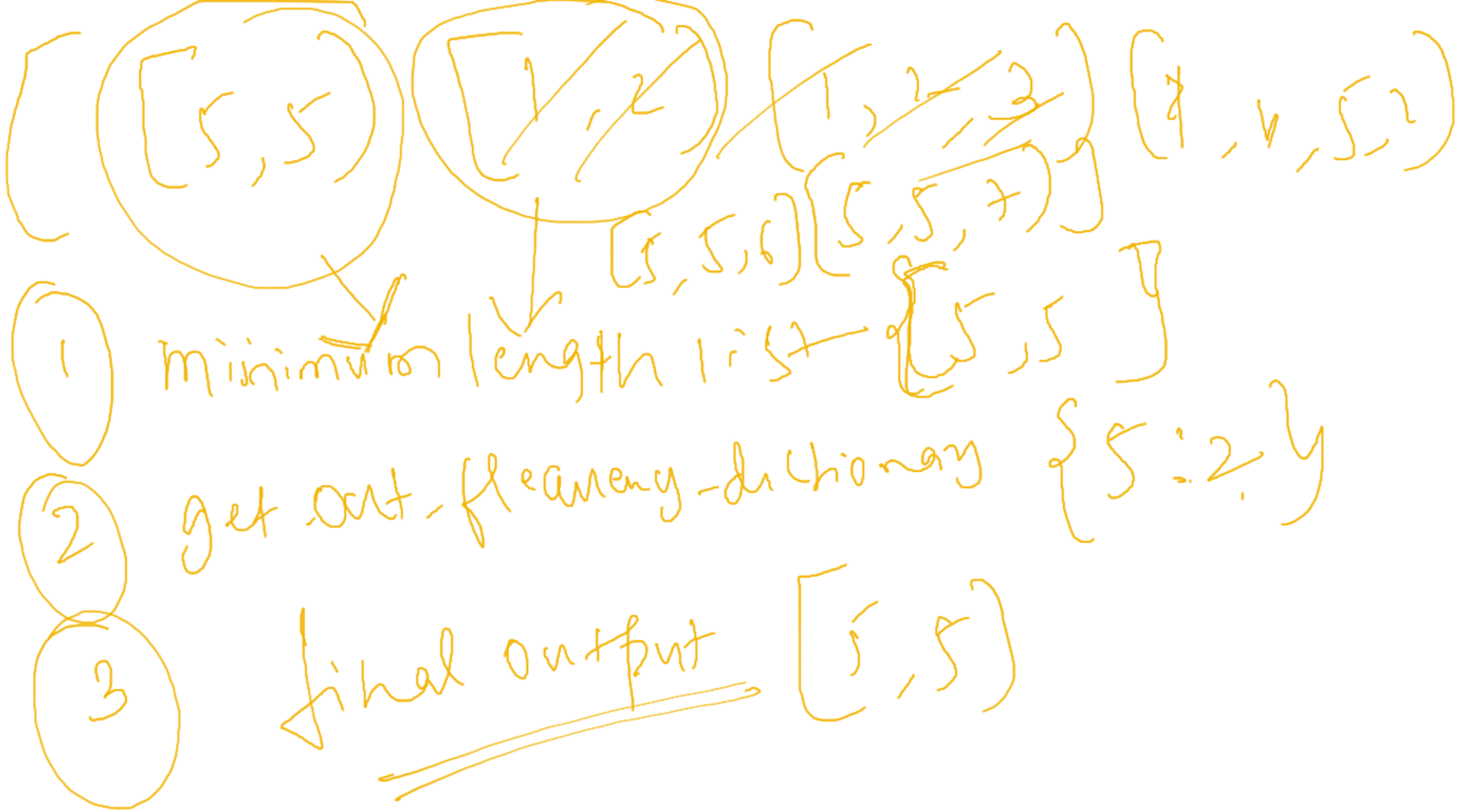

[5,5,10] [5,8,10] [5,5,10,2]
[5,5,5,10]
first step
5,10
2nd
5:2, 10:1
Counter = 0
5:2
County = 1
If County: 2
key will be Common elements in the list
Value will be frequency of the Common element



improve your code


counter = 0
for
counter =
1st step
element
frequency
counter


[[5,5] [3,5,6]]
Smallest_list
min_length_dict
[5,5]
{5:2}
dict
{k:v}
value
dict[k]:2
{5:1}
modified
Smallest element = 5
list_ = [3,5,6]
Counter = ?
Counter = 1
try
dict[Smallest_elem] ?
2 > Counter
dict[Smallest_elem] = Counter


1 minimum length list
2 get out_freaneny_dictionary
3 final output

[[5,5,10],[5,5,7,8],[5,8,9,10]]

1. Smallest list = [5,5,10]

2. final_out_dictionary = {5:1, 10:0}

3. finalOutput = [5]