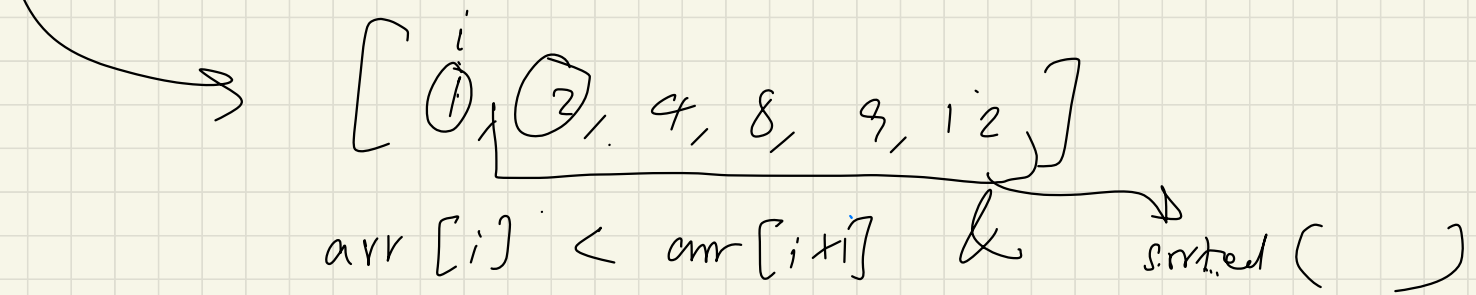
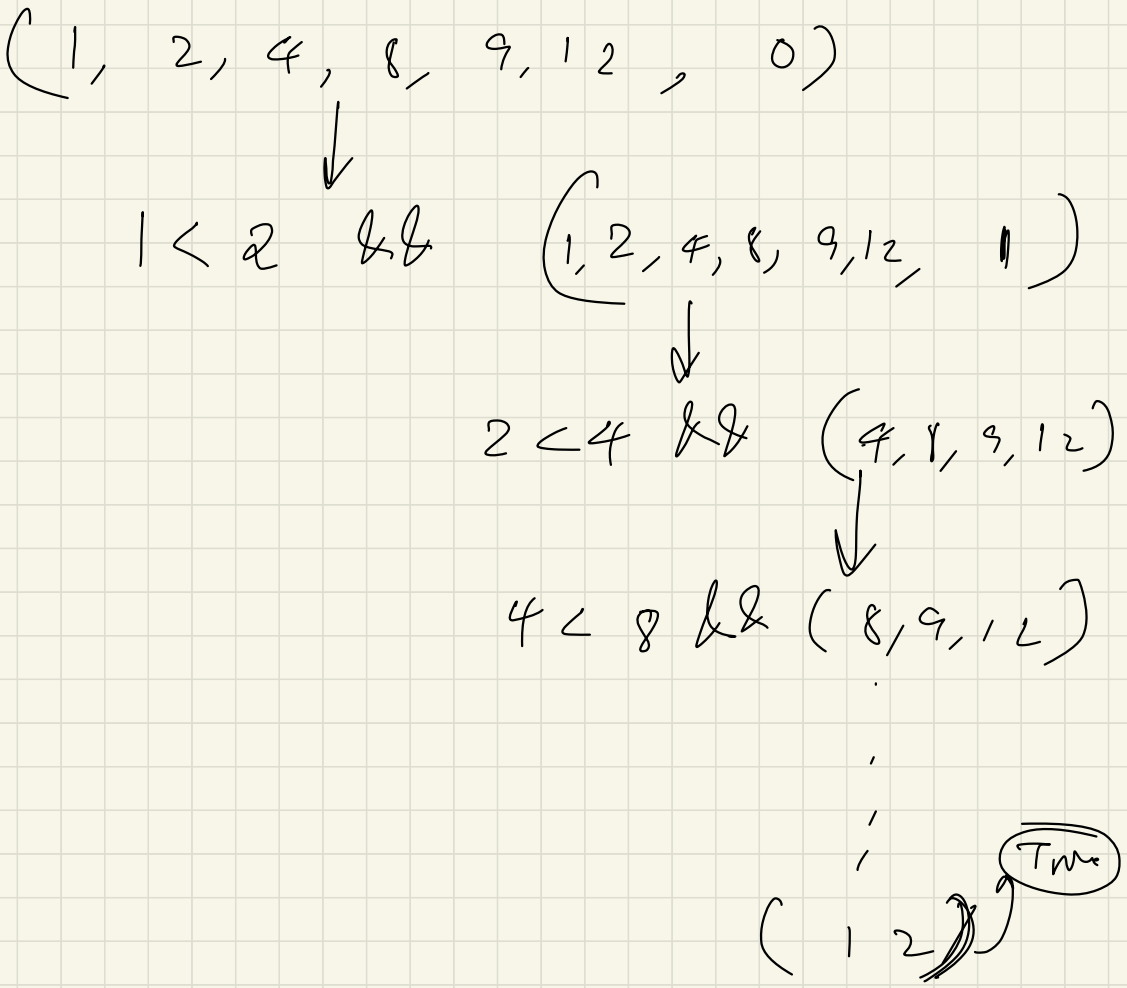
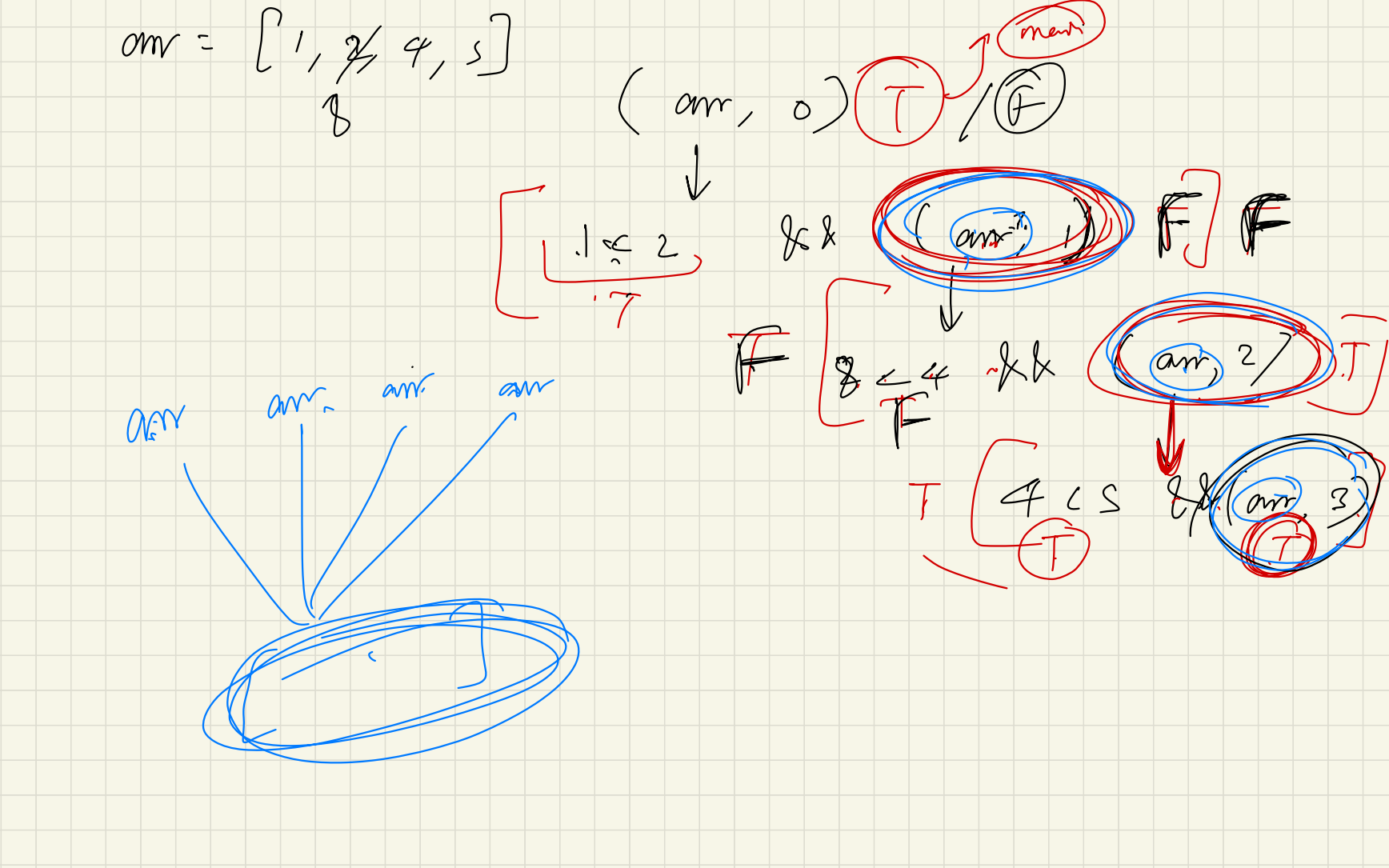
Q: Find if array is sorted or not.

arr = [1, 2, 4, 8, 9, 12]

arr = [1, 2, 4, 3, 8, 9]




(1, 2, 4, 8, 9, 12, 0)
1 < 2 &&
(1, 2, 4, 8, 9, 12, 1)
2 < 4 &&
(4, 8, 9, 12)
4 < 8 &&
(8, 9, 12)
(12)
TLE


arr = [1, 2, 4, 5]
(arr, 0)
T
F
mem
1 <= 2
T
&&
(arr, 1)
F
F
F
2 <= 4
F
&&
(arr, 2)
T
T
4 <= 5
T
&&
(arr, 3)
T
arr
arr
arr
arr

Q:

arr = [3, 2, 1, 18, 9], target = 18

(arr, target, 0)

if arr[0] = target || (arr, target, 1)

T

arr[1] = target || (arr, target, 2)

:

using binary search

Q: arr = [1, 2, 3, 4, 4, 8] , (target = 4)

ans = [3, 4]

fun(arr, target, index, list)

if (index = end) {

return list

}

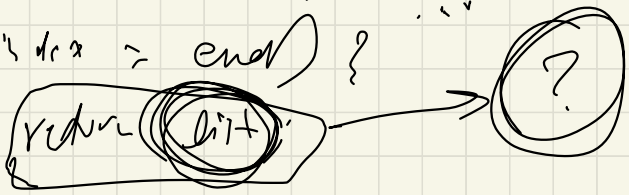
if (target = arr[index]

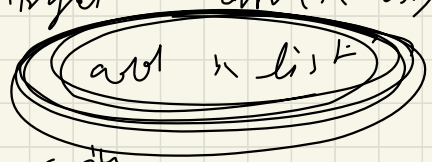
add index to list

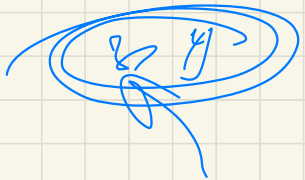
keep going

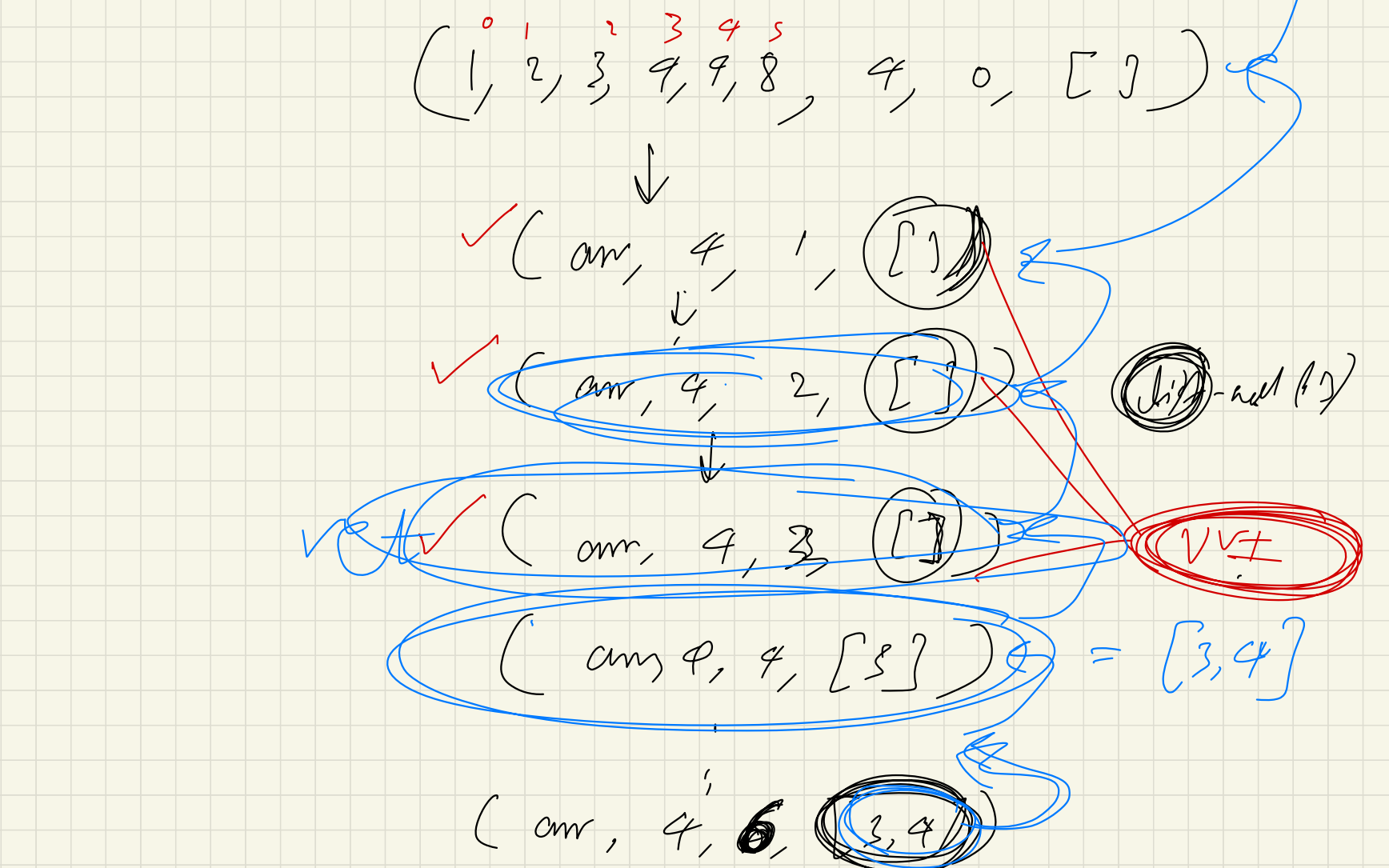
rec. fun(

}








0 1 2 3 4 5
(1, 2, 3, 4, 9, 8, 4, 0, [])
(arr, 4, 1, [])
(arr, 4, 2, [])
(arr, 4, 3, [])
(arr, 4, [3])
= [3, 4]
(arr, 4, 6, [3, 4])

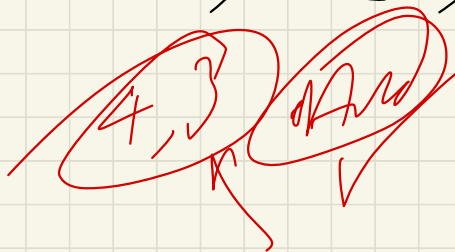
Q: V v I : Goal : ❖ return the list

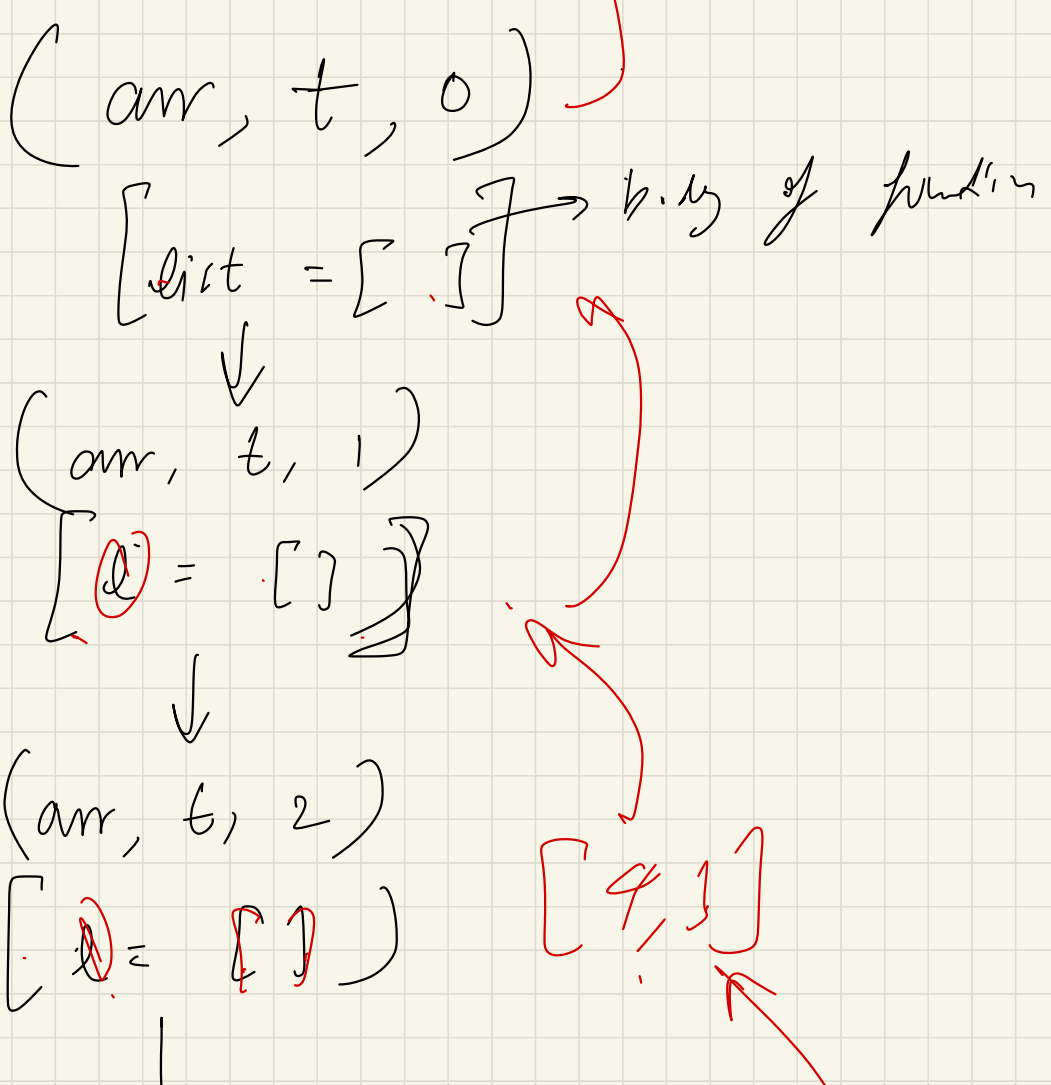
❖ dont take it in argument

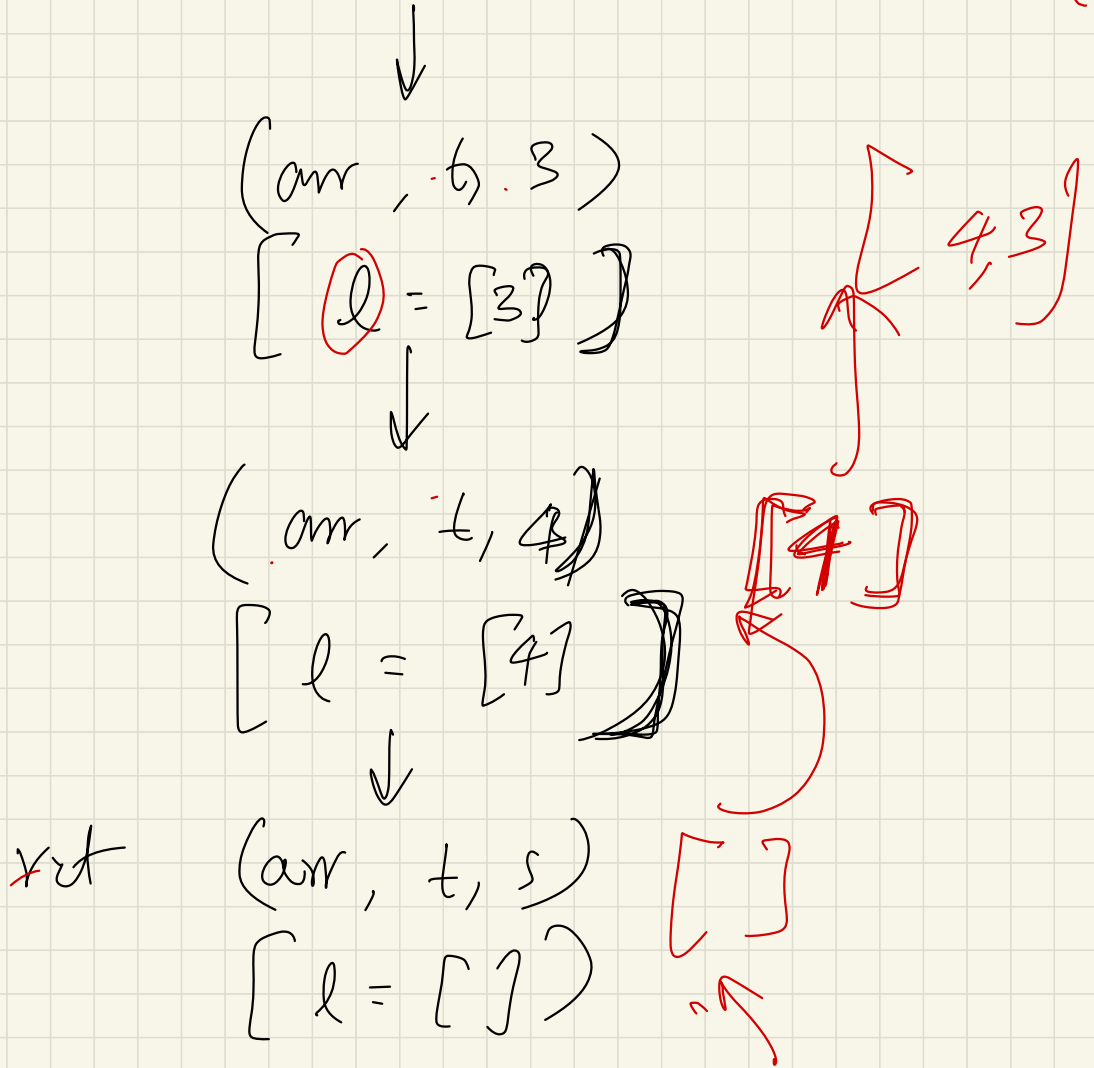
Challenges : ❖ rt will be A

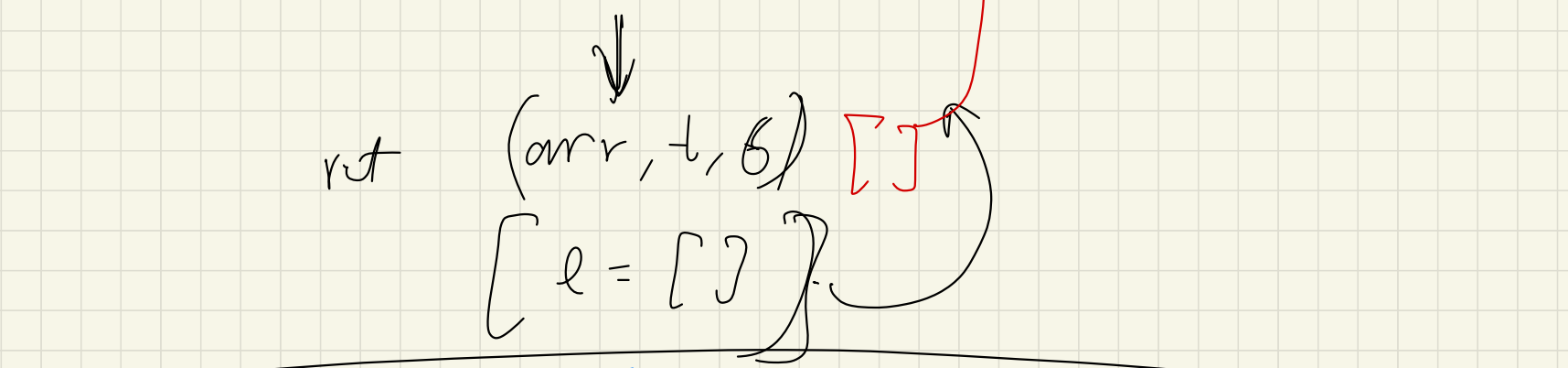
Problem : Every call will have a new list.

arr = [1, 2, 3, 4, 4, 8], target = 4,




(arr, t, 0)
[dict = [ ]]
b.by of function
(arr, t, 1)
[d = [ ]]
(arr, t, 2)
[d = [ ]]
[4, 3]


(arr, t, 3)
[l = [3]]
(arr, t, 4)
[l = [4]]
ret
(arr, t, 5)
[l = [ ]]
[4]
[4,3]
[ ]



Q:

arr = [5, 6, 7, 8, 9, 1, 2, 3]

indices: 0 1 2 3 4 5 6 7

s = 0, e = 7, m = 3

t = 7

① if arr[s] ≤ arr[mid]

if key ≥ arr(s) & ≤ arr(mid)

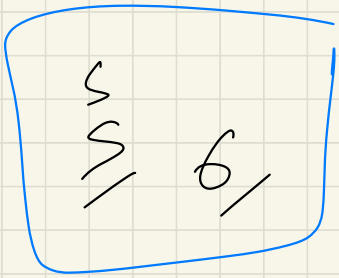
end = m - 1

else

s = m + 1

(2) if key ≥ arr[m] && ≤ arr[e]

s = m+1



(6)

(3) else

e = m − 1