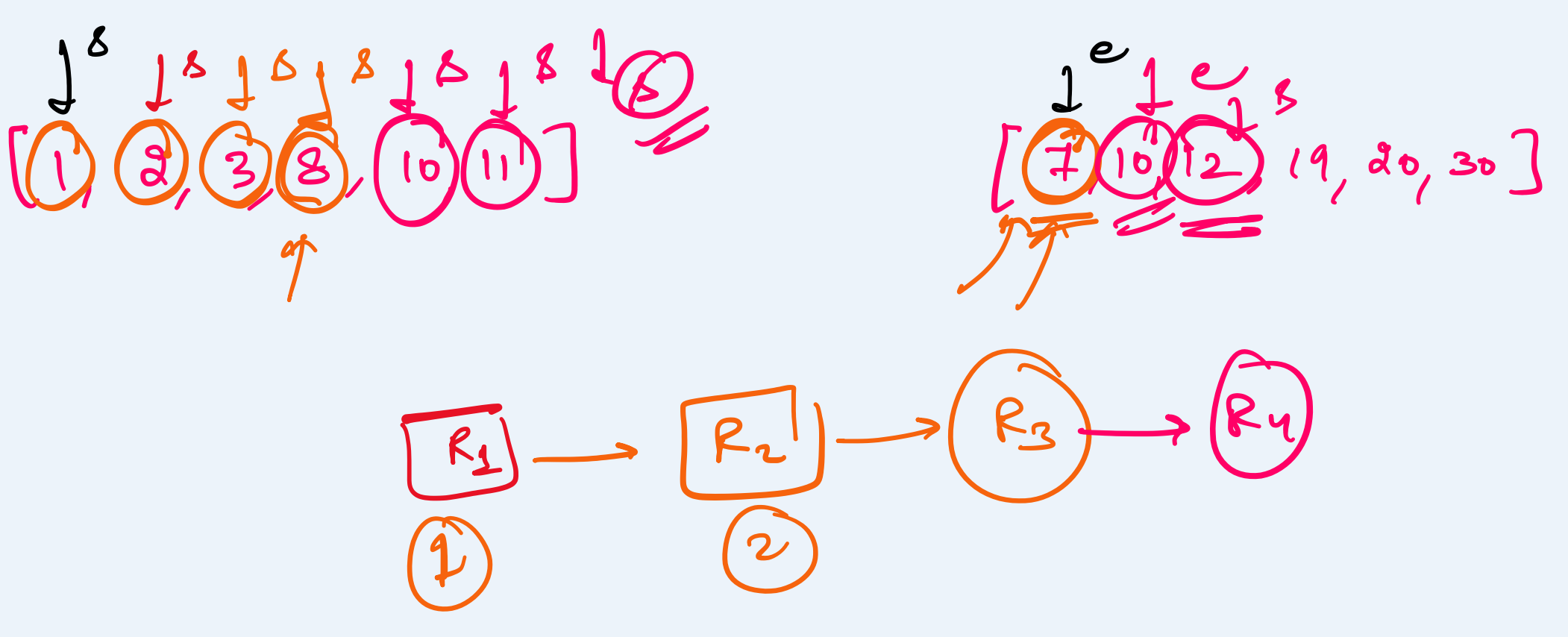
Q Meeting rooms

intervals : [1, 10], [2, 7], [3, 19], [8, 12], [10, 20], [11, 30]



Q Maximize sum of array after k negations.

[2, -3, -1, 5, -4] , k = 2

output → 13

1. -4 → 4

-ve → positive

[2, -3, -1, 5, 4]

2. -3 → 3

[2, 3, -1, 5, 4]

→ 2 + 3 - 1 + 5 + 4 = 13

[2, 3, 4, 5] → Pick min ele

x [-2]

k → even

2 → -2 → 2

1. Sort the array
2. (First ele == pos / neg)
   1. If k → even, dont do anything
   2. else modify the min ele.
3. mark -ve ele positive for given k.
4. After k is still not even, mark the lowest ele → -ve.

Q Job Scheduling

Each job has a deadline and a profit associated.

Eg:

Job	Deadline	Profit
a	4	20
b	1	10
c	1	40
d	1	30

c	a		
1	2	3	4

T profit = 60
40 + 20

- → 1 job = 1 day
- → Job can be performed in any order
- → Not necessary to perform all jobs
- → maximize profit.

Eg 2

Job	d	p
a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

ans → 142

c	a	e

→ 100 + 27 + 15

let's say,

a		500
b		100
c		20

choose the job with max profit

a	2	100
b	1	100
c	3	100

what should be the ordering here ??

a | b | c (3 options)

c 1 option

For day 1, I have all the options.

* **Always perform a job on its deadline day.**

1. which job to choose first ??
   → offers max profit
2. when to prefer a job ??
   → as close to the deadline as possible.

1. Sort the jobs as per profit.
2. Try to find a slot

J	d	p
a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

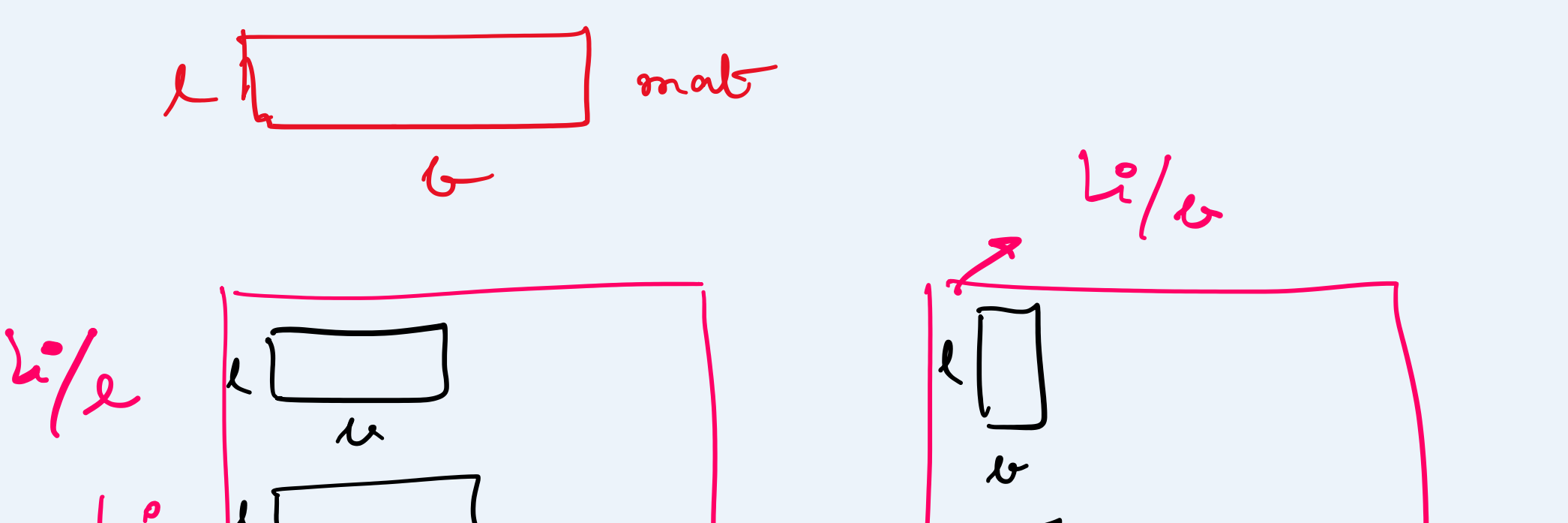
→

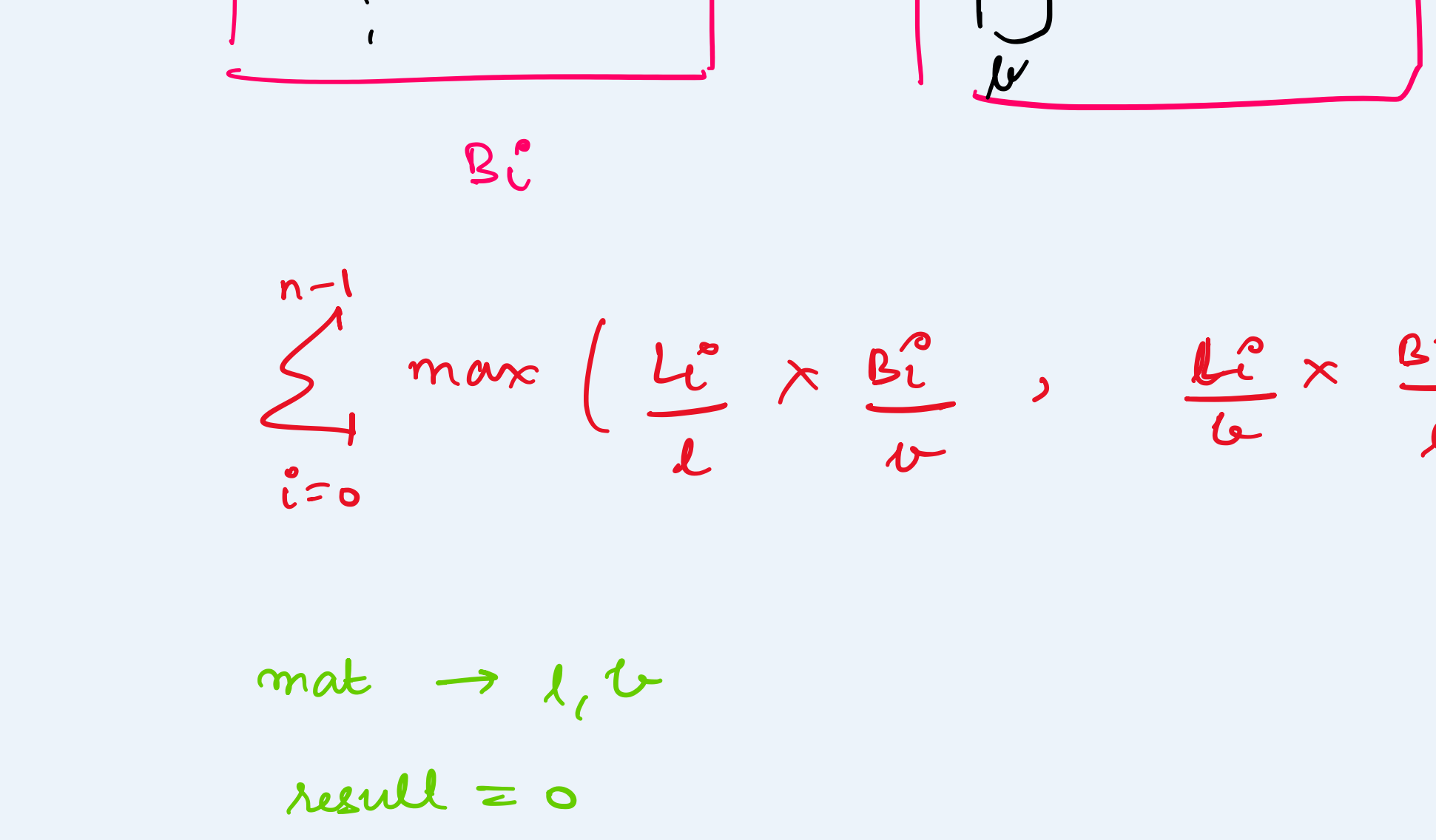
J	d	p
a	2	100
c	2	27
d	1	25
b	1	19
e	3	15

c	a	e
1	2	3

→ 100 + 27 + 15
→ 142

Q RAMDEV (Codechef)





$$\sum_{i=0}^{n-1} \max\left(\frac{L_i}{l} \times \frac{B_i}{b}, \frac{L_i}{b} \times \frac{B_i}{l}\right)$$

```
mat → l, b
result = 0
for (i=0; i<n; i++) {
        int Li = input();
        int Bi = input();

    result += max(((Li/l)*(Bi/b)), ((Li/b)*(Bi/l))
```