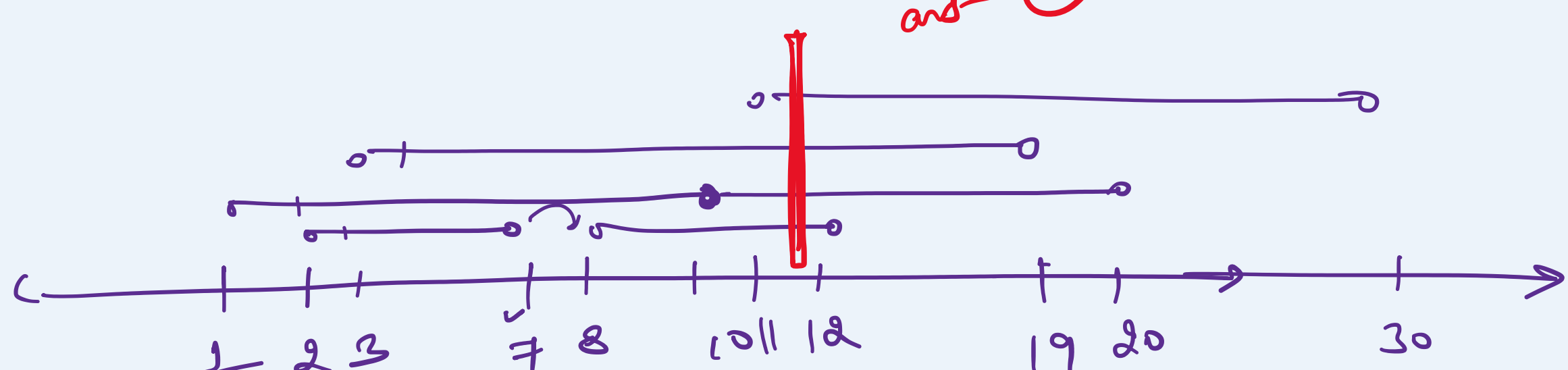
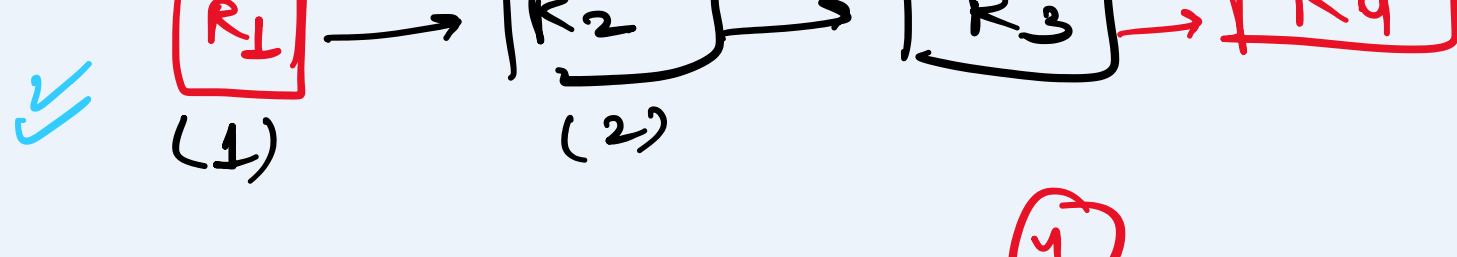
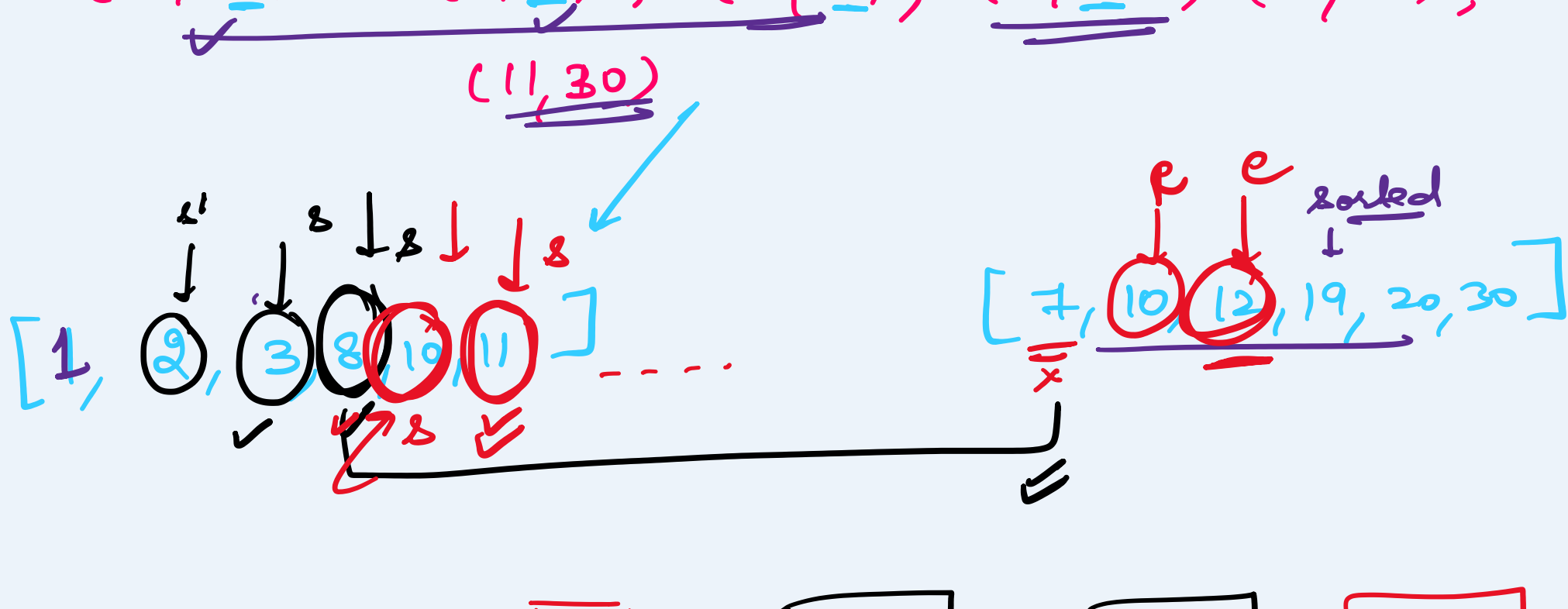
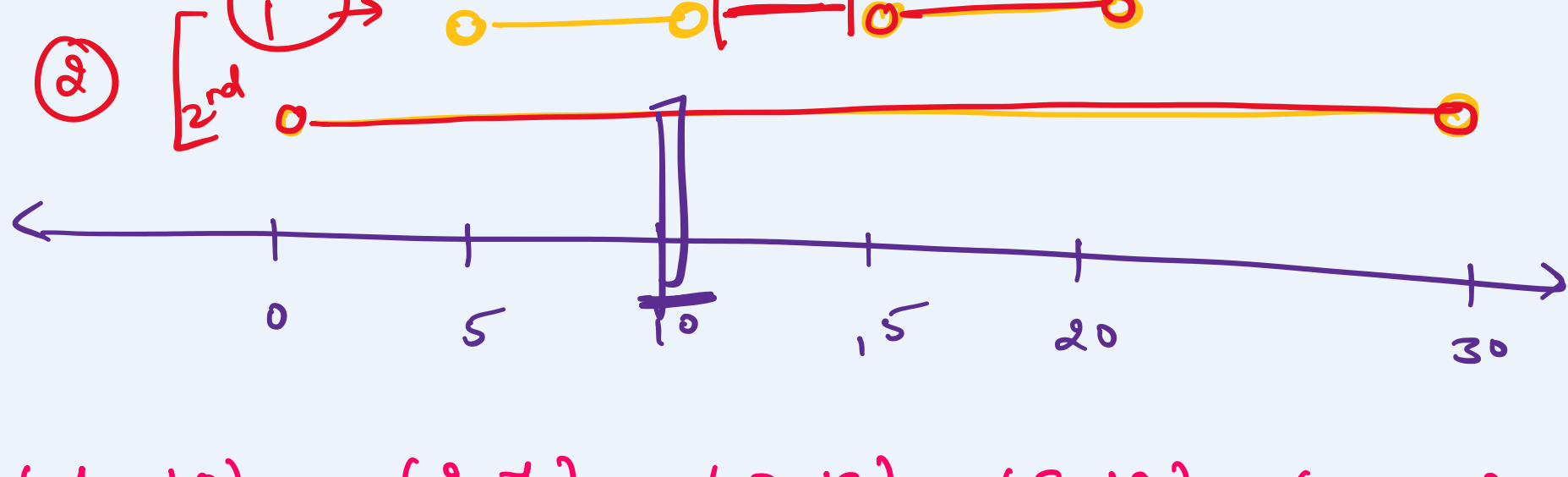
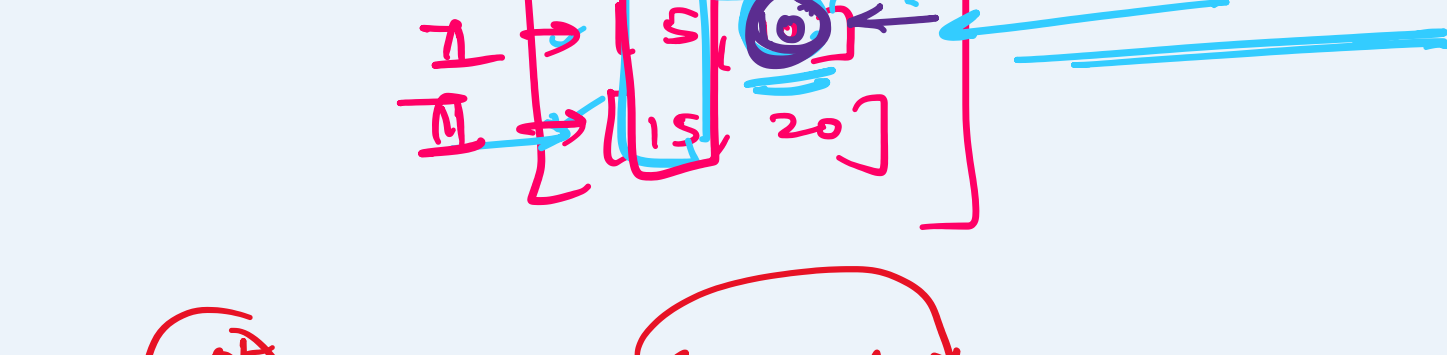


Q Meeting rooms

→ Intervals → [start, end]

min no. of conference rooms required.



Q Maximum sum of array after k negations

✓ k operations

✓ [3, -1, 0, 2] k=3

[3, 1, 0, 2]

[3, 1, 0, 2]

[3, 1, 0, 2] → ans → 6

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

① -4 → 4

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

② -3 → 3

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

2 + 3 - 1 + 5 + 4 = 13

[2, 1, 1]

k=2

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

k=3

k=1

① sort the array

② (first ele == pos/neg)

① if (k == even), don't do anything

② Else modify the min ele.

③ mark -ve ele positive for given k.

④ After that 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

T → 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 40 + 20

→ 1 job ≈ 1 day

→ Job can be performed in any order

→ Not necessary to perform all jobs

→ maximum profit

Job d p

a 2 100

b 1 10

c 2 27

d 1 25

e 3 15

→ 27 + 100 + 15

→ 142

let's say

a 1 50

b 1 100

c 1 20

choose the one with max profit

a 2 100

b 1 100

c 3 100

what should be the ordering here??

→ 3 options

→ 1 option

for day 1, I have all options.

* Always the perform the job on its deadline day.

① which job to choose first??

→ offers max profit

② when to prefer a job??

→ as close to its deadline as possible.

① sort the jobs as per profit

② try to find a slot

→ a 2 100

→ b 1 100

→ c 2 27

→ d 1 25

→ e 3 15

→ 27 + 100 + 25

→ 142