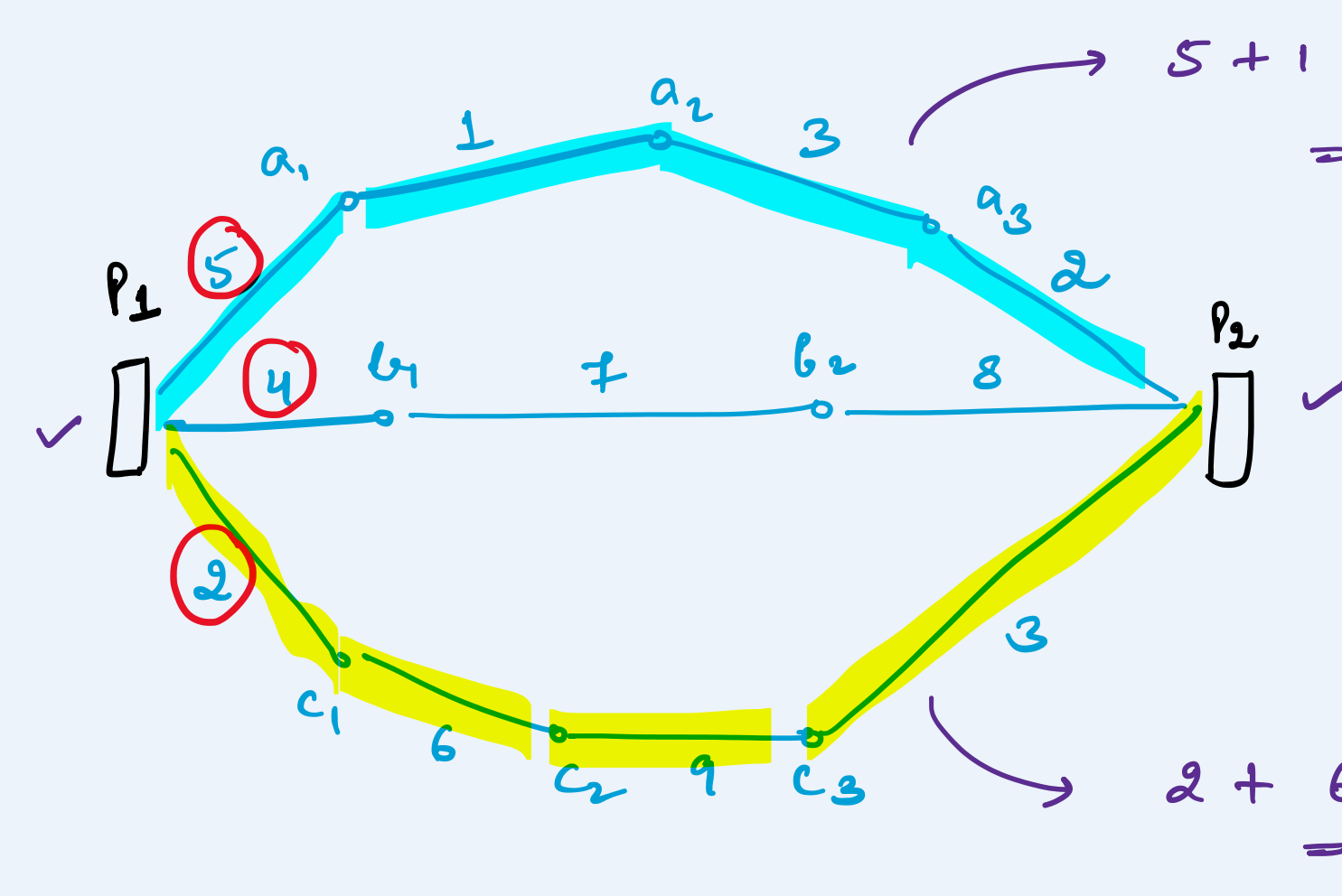
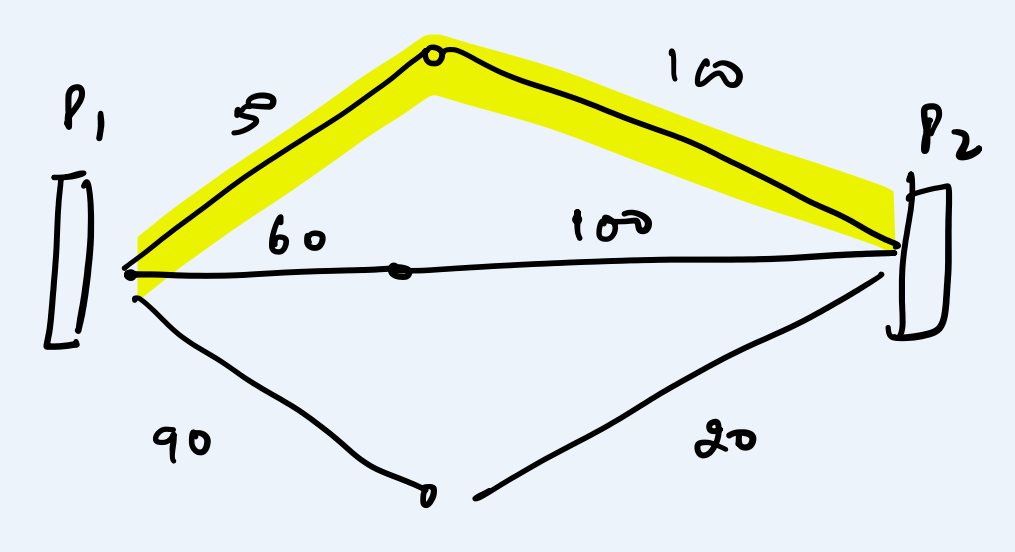
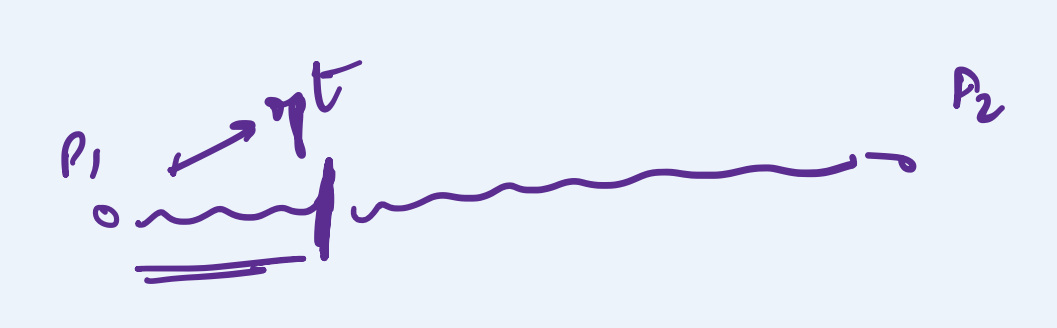
Monday, 25 October 2021 11:52 AM

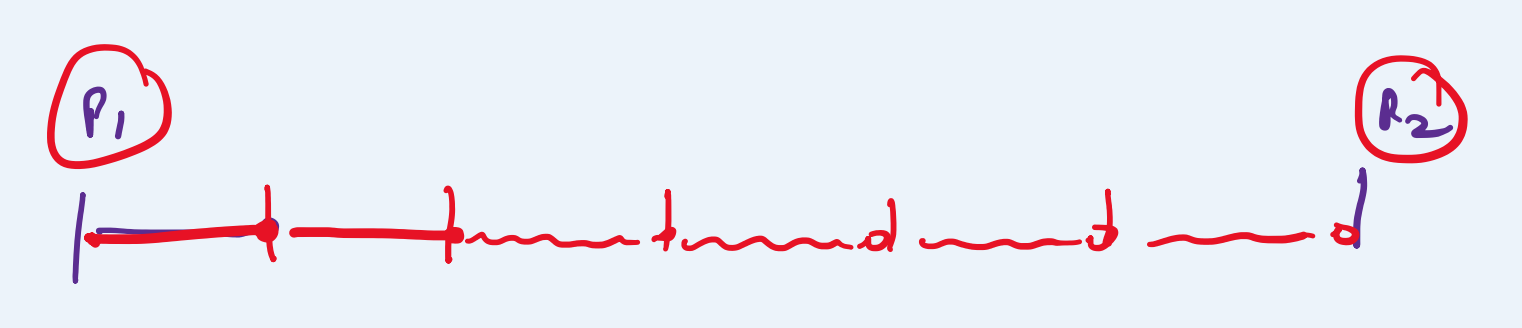
Greedy Algorithms

Greedy Algorithms always make the choice that seems to be best at the moment without taking into consideration the future consequences.









1. Greedy algorithms are always based on intuition & observation.
2. Greedy algorithms can only work on when local answer contributes to the global answer.

Two properties:

1. greedy choice property
   if by choosing the best choice at each step without reconsidering the previous step once chosen.

2. optimal substructure
   if the optimal overall solution to the problem corresponds to the optimal solution to its subproblems.

Q Task schedules (leetcode)

Ans Total no. of CPU cycles = Busy + Idle → ??

No. of busy slots = no. of tasks execute = len(arr)

Max no. of idle slots is defined by the freq of most freq task.

A B A A B C A A, n = 2 A → 5

A _ _ A _ _ A _ _ A _ _ A

f_max = 5

Max possible idle time = (f_max − 1) × cooling period

= (5 − 1) * 2 = 8

Execute other tasks in idle time.

→ A B A A B C A A

1. Freq(
   → A → 5
   → B → 2
   → C → 1

A B C A B _ A _ _ A _ _ A

8 − 2 − 1 → 5

Total time → 8 + 5 → 13

Q Given an array of +ve & −ve integers, find the min - possible subset product.

Eg: {−1, −1, −2, 4, 3} ans = −24

Eg: {−1, 0} ans = −1

Eg: {−1, 0, 2} ans = −2

what if all +ve no ?? [no negative no]

[2 1 3 5]

[2 1 3 0 4 5]

ans → min ele of the array.

what if 1 −ve no, is present ??

[2 1 3 0 4 5 −1]

ans = −1 × all other no except 0.

what if 2 −ve no present ??

[2 1 3 2 4 5 −2 −1]

→ ans = −2 (−ve with smaller magnitude) × all positive +ve except 0.

Defining factor → no of negative elements.

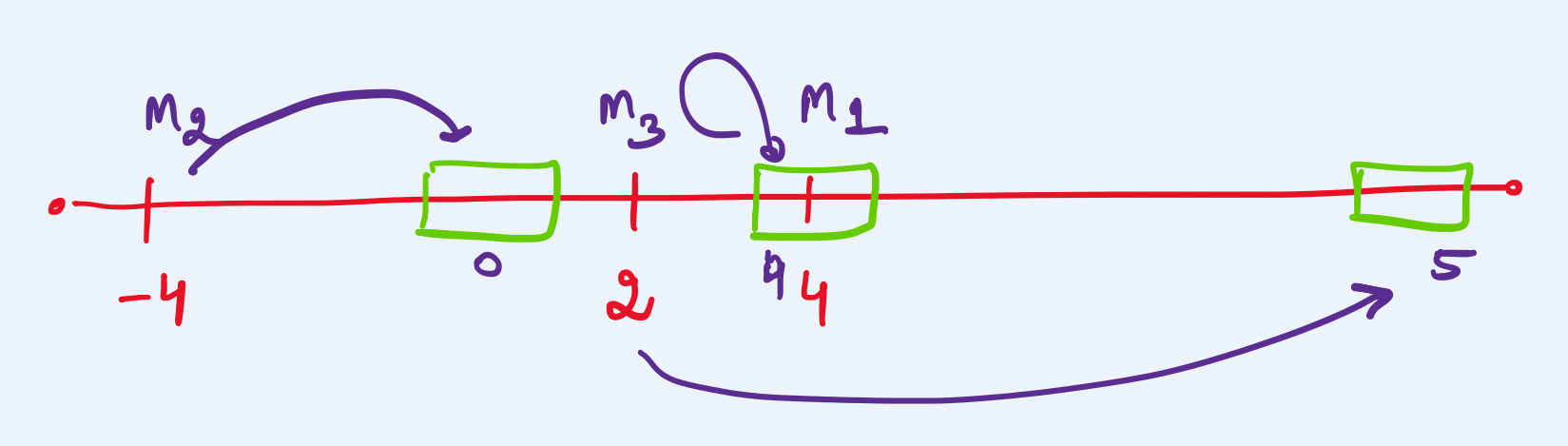
Cases:

1. no negative no → min(arr)
2. even no. of negative no.
   → Π[a[i]] × negative no. of min value
   → a[i] ≠ 0
3. odd no. of negative no.
   Π[a[i]]
   a[i] ≠ 0

Q Mice and holes.

mice = [4, −4, 2]

holes = [4, 0, 5]



Ans depends on the max distance that any mice has to travel.

1. Sort the mice positions
2. sort hole positions.

[mice[i] − hole[i]]

mice → [−4, 2, 4]

holes → [0, 4, 5]

4 2 1

i = 0 = |−4 − 0| = 4

i = 1 = |4 − 2| = 2

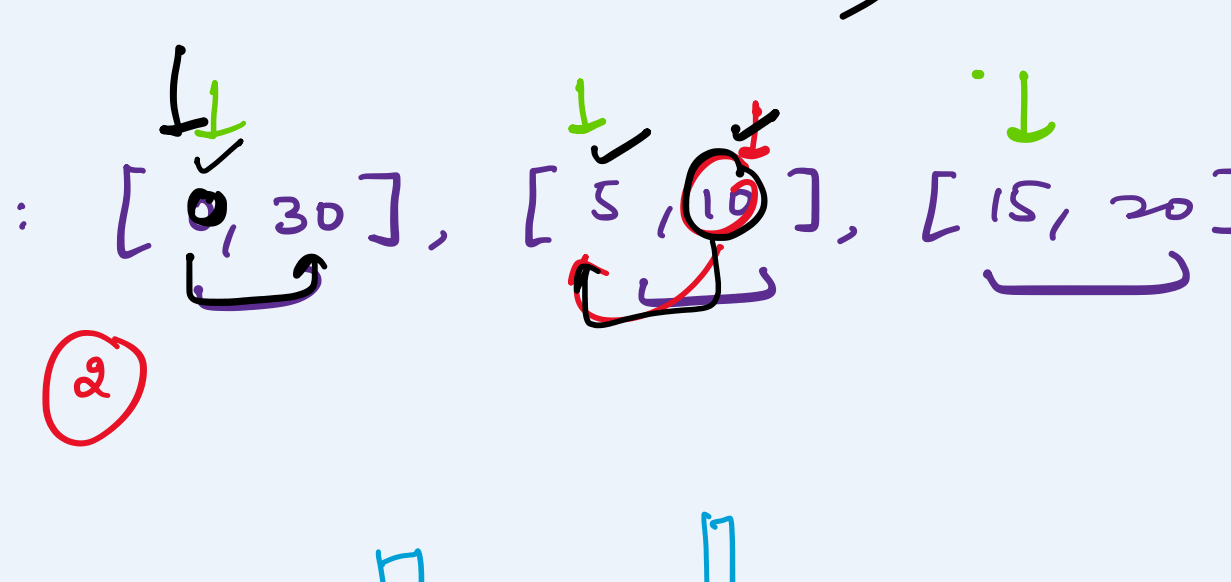
i = 2 = |4 − 5| = 1

max(4, 2, 1) = 4 ans

Q Meeting rooms II (leetcode)

intervals: [[0, 30], [5, 10], [15, 20]]

ans → 2



ans → 2

(1, 10), (2, 7), (3, 19), (8, 12), (10, 20), (11, 30)

start time [1, 2, 3, 8, 10, 11]

end time [7, 10, 12, 19, 20, 30]

R1 → R2 → R3 → R4

[if start[s_ptr] < end[e_ptr] need a new meeting room.]