Smallest Range Covering Elements from K Lists

Saturday, 18 September 2021 10:31 AM

all lists are sorted

[4,	10,	15,	24,	26],
[0,	9,	12,	20],	
[5,	18,	22,	30]	

priority Queue = min

~~min~~ min will move ahead to increase the possibility of shrinkage of Smallest Range for result.

max = ~~5~~ 9

Modulo management

$(a+b)\%mod = (a\%mod + b\%mod)\%mod$

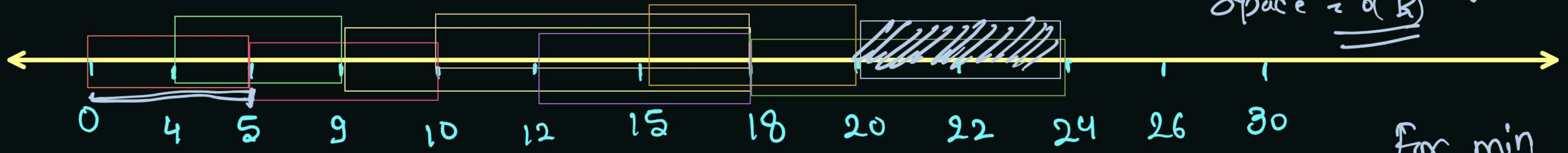
$(a-b)\%mod = (a\%mod - b\%mod)\%mod$

$(a*b)\%mod = [(a\%mod) * (b\%mod)]\%mod$

~~$(a/b)\%mod = [(a\%mod)/(b\%mod)]$~~

if modulo operation is required, take variable as 'long' type

time - O(n log k)
Space = O(k)



smallest = ~~0~~ ~~4~~ ~~5~~ ~~9~~ ~~10~~ ~~12~~ ~~15~~ ~~18~~ 20

largest = ~~5~~ ~~9~~ ~~10~~ ~~18~~ ~~18~~ ~~18~~ ~~20~~ 24 e

min length = ~~∞~~ ~~5~~ 4

st ~~10~~ 20
b ~~25~~ 24

For min use priority queue and for max use a max variable and manage in Running time.

Car Pooling

Saturday, 18 September 2021 10:31 AM

count st. Ending.

[3,2,7]

[3,7,9]

[8,3,9]

Capacity = 11

car is moving is some direction from start to End.

Given coordintes of starting and Ending for ride and no. of passengs in that ride.

Given a capacity of passenger in vehicle.

Is is possible to pool all rides

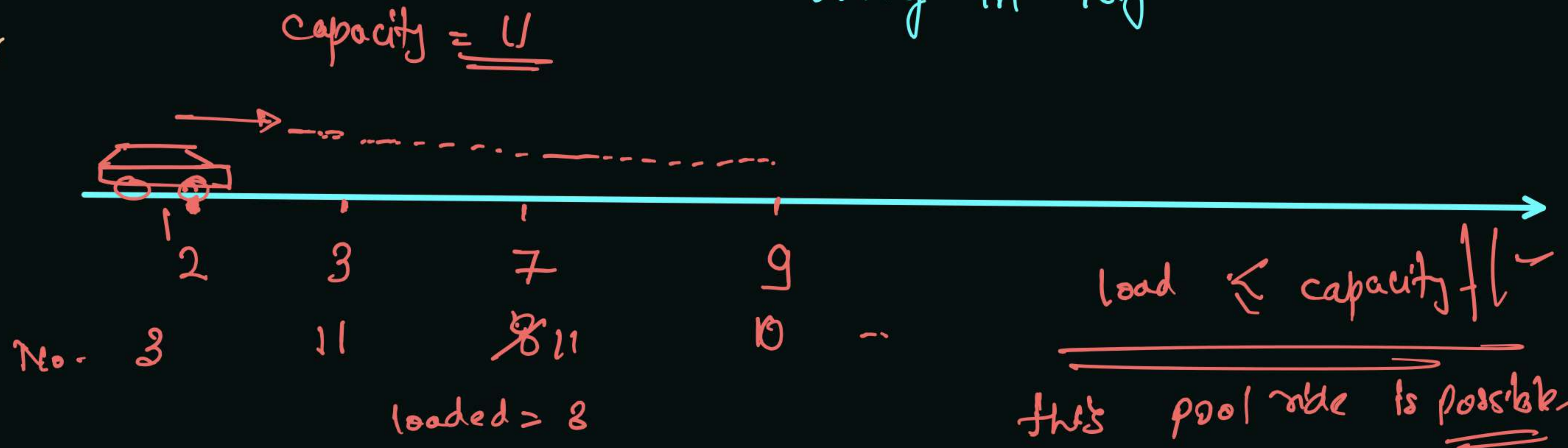
Highway In Right direction, & vehicle is also moving In Right direction.

rows → ride

column → 0 → No. of passenger

column → 1 ⇒ starting of Ride

column → 2 → Ending of Ride



load ≤ capacity 11 ✓

this pool ride is possible.

ride	Starting	Ending	Passenger
	2 ⟷	6 →	3
	3 ⟷	7 →	2
	7 ⟷	11 →	2
	6 ⟷	8 →	3
	11 ⟷	15 →	1
	9 ⟷	12 →	6
	8 ⟷	10 →	4 ✓

endif1 = -val

Capacity = 20

Range Addition

change logic → ending → Decrement

2	4	5	6
3	3	3	-3

Pool Ride is possible

True

2	3	6	7	8	9	10	11	12	15
3	~~3~~	~~3~~	~~2~~	~~5~~	~~2~~	~~5~~	~~5~~	~~1~~	1
	5	~~5~~	~~7~~	~~8~~	~~11~~	~~11~~	~~6~~	7	
		8	10	12	15	15	12		

No. of passenger < Cap

Maximum Product Subarray

Saturday, 18 September 2021 10:31 AM

Maximum Sum Subarray → Kadane's Algo.

break point ⇒ '0' is break (L.P.) 0 | -ve (R.P.) → single -ve.

L.P R.P

Ex →	[1	-2	4	-3]	0	[13	-3	1]	0	[4	-1	4	-3]
left product →	1	-2	-8	24	1 R84	13	-39	-39	1	4	-4	-16	48
right product	24	24	-12	-3	1	-39	-3	1	1	48	12	-12	-3

Subarray

product →

num * 0 = 0

Max in left product and right product is final result

+ve * +ve = +ve

+ve * -ve = -ve

-ve * +ve = -ve

-ve * -ve = +ve

1 -2 8 →

1 -2 -16

← -16 -16 8 → max

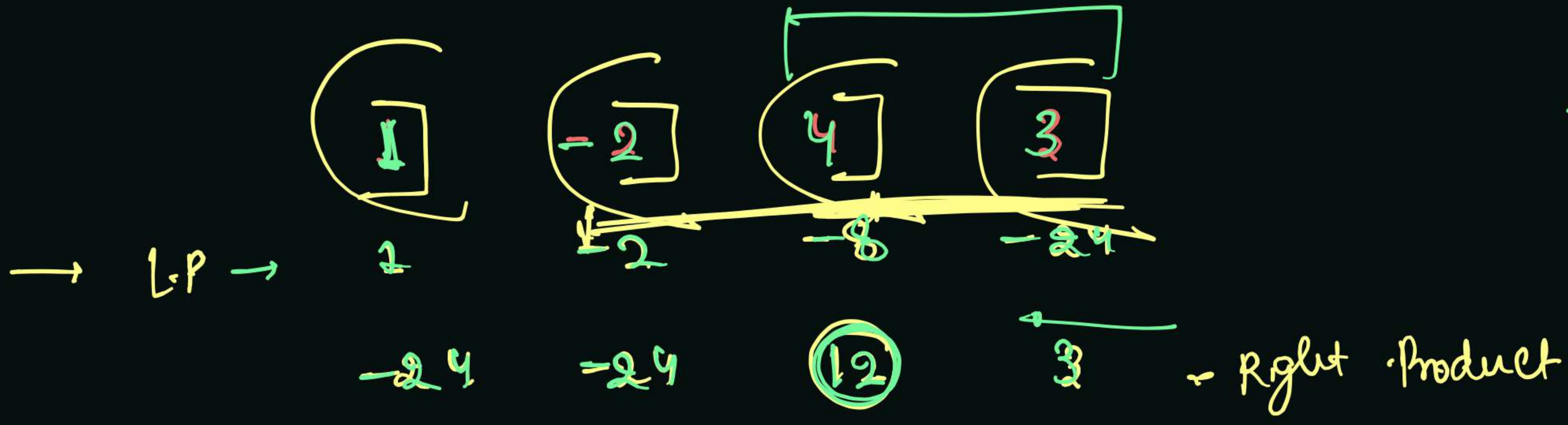
SORRY ☺

Never Including in Max. product subarray. Except this →

product

1 0 -2

Result = 0



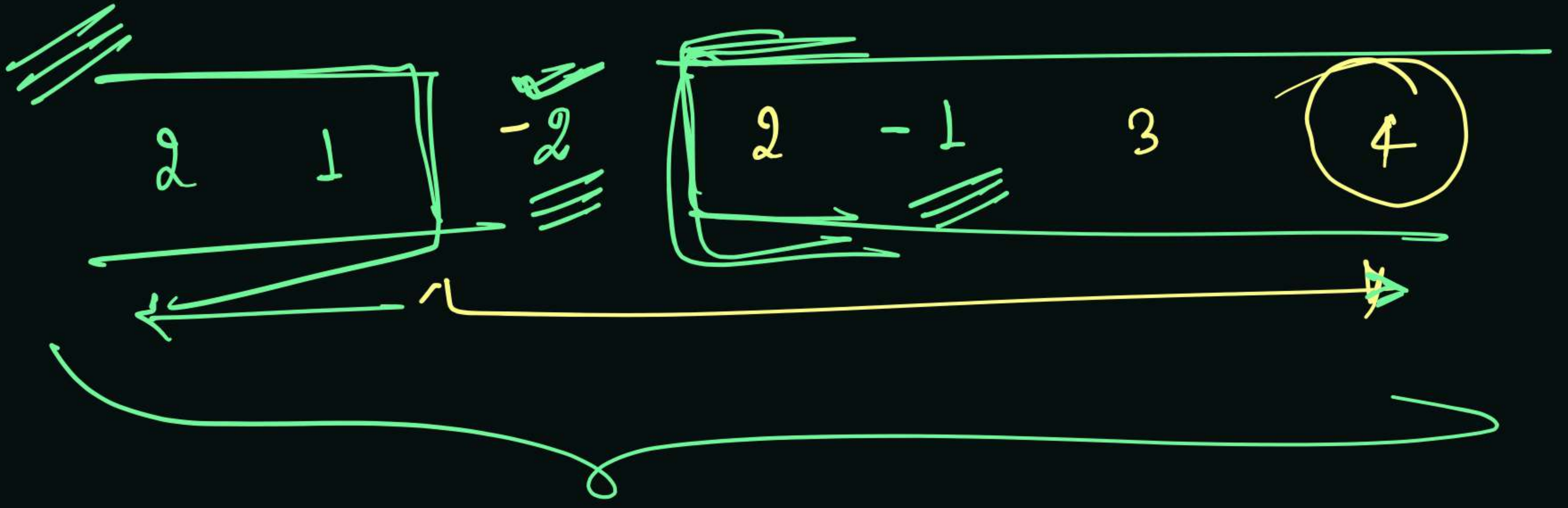
-ve * -ve = +ve

-ve * +ve, => +ve

+ve * -ve = -ve

+ve * -ve =

L.P. Sub.. ar



-ve → marker

└ separater

↓

It can be

helpfull

Kadane's → senior

max sum → 3 days