K- Concatenation

Friday, 29 October 2021 6:15 PM

Application of Kadane's

→ array → [1 2 3], k = 3

→ concatenate array k times, final Array → [1 2 3 1 2 3 1 2 3]

output - Max. Sum of Subarray after k-concatenation.

constraints

- Range of k → 10^5
- size of single array → 10^5
- is it possible to concatenate array?? size - 10^{10}] X

circular loop k-times →

Time limit = 1 Sec.

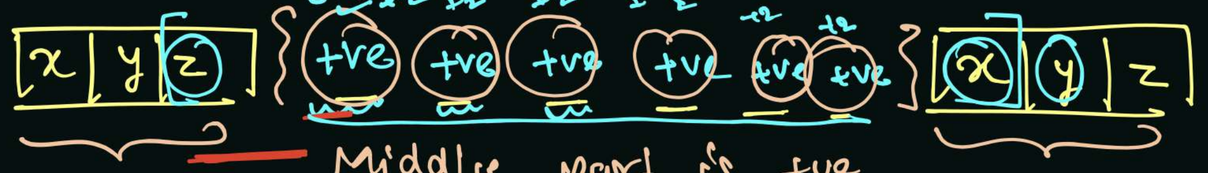
in 1 sec. we can do at max 10^9 operation.

we have 10^{10} operations (max), so to avoid TLE we not apply circular loop.

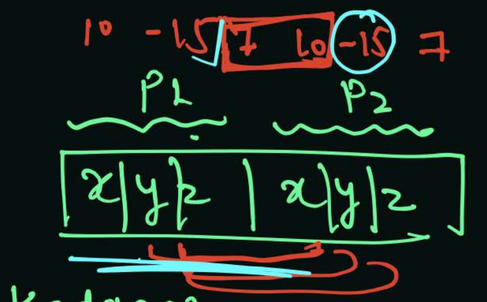
K=1 → Kadanes (Normal) array → [x | y | z]

① Find total sum, Tsum

case-I if (Tsum >= 0) Tsum = tve , = x+y+z = tve

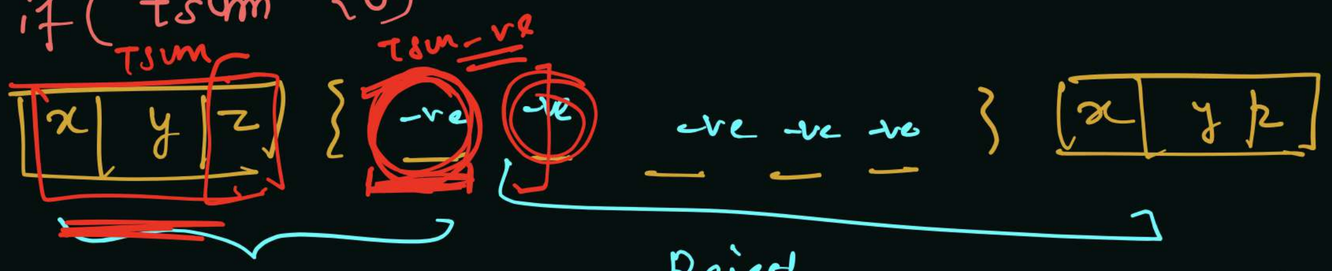


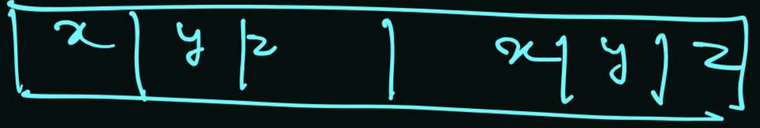
⇒



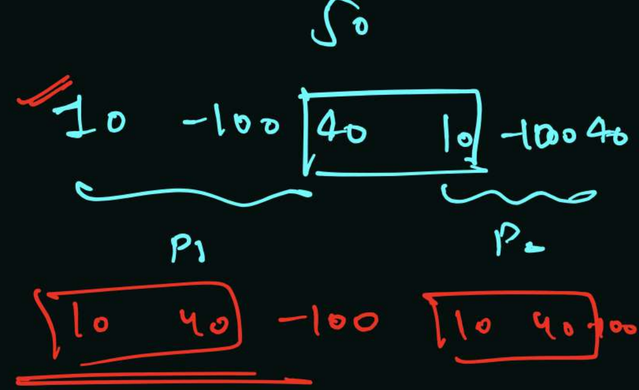
result = Kadanes of (1, 2) + [sum*(k-2)]

case-II if (Tsum < 0)





Ex



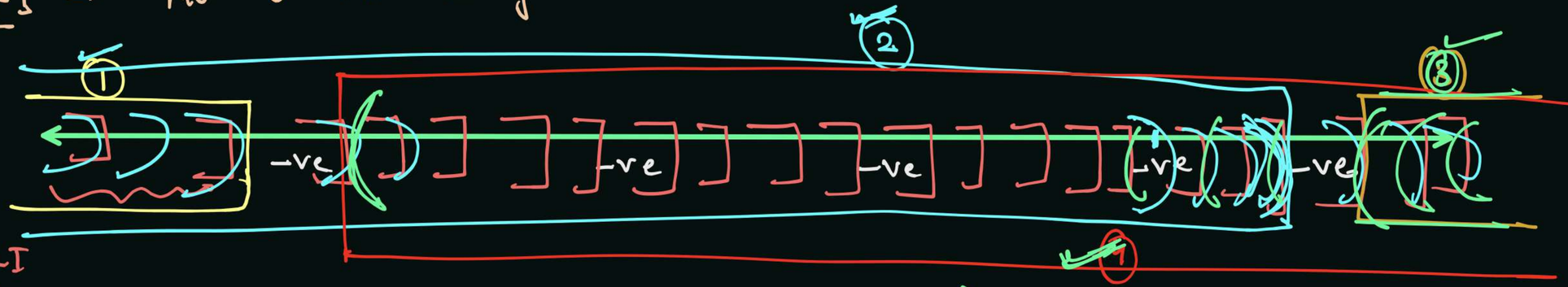
result = Kadanes of (1, 2) + 0

Max Product Subarray

Friday, 29 October 2021 7:31 PM

this is different from kadane's

case-I → No '0' in array →



Iteration-I

left product with maximisation of result

Iteration-II

right product with maximisation of result

↳ can we skip -ve number??

odd no. of -ve → single -ve skip

Even no of -ve → no need of skip it

case-II → If zero is present → Reset cmax

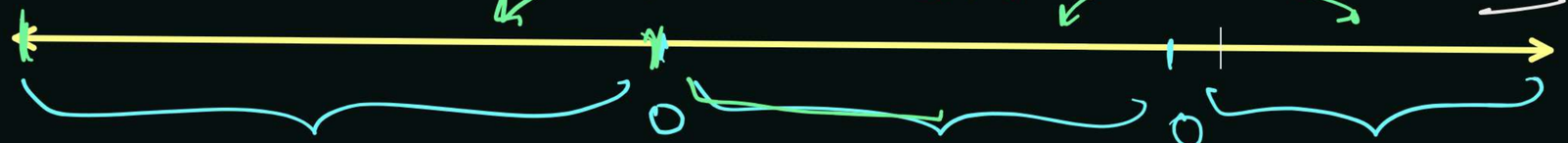
split in sections

why →

-ve * -ve → +ve

-ve * +ve → -ve

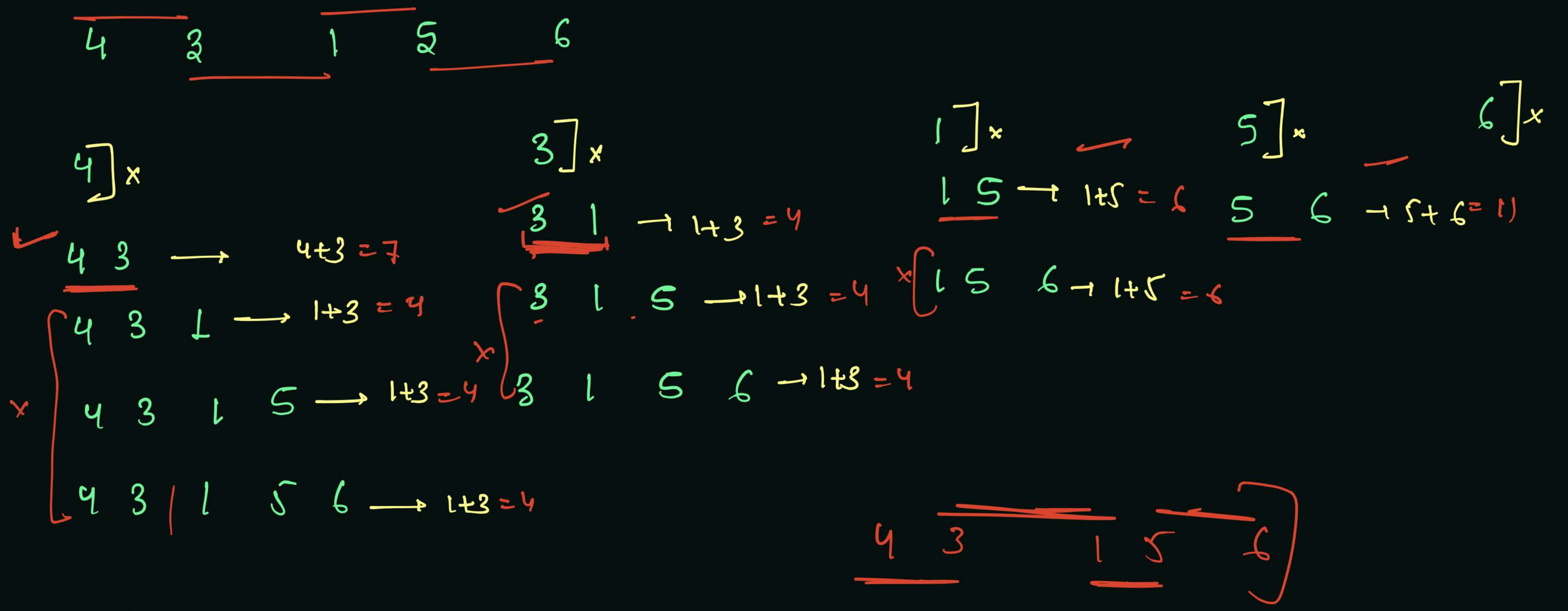
+ve * +ve → +ve



if(arr[i] == 0) cmax = 1

Maximum Sum Of Smallest and Second Smallest from all subarrays

Friday, 29 October 2021 8:23 PM



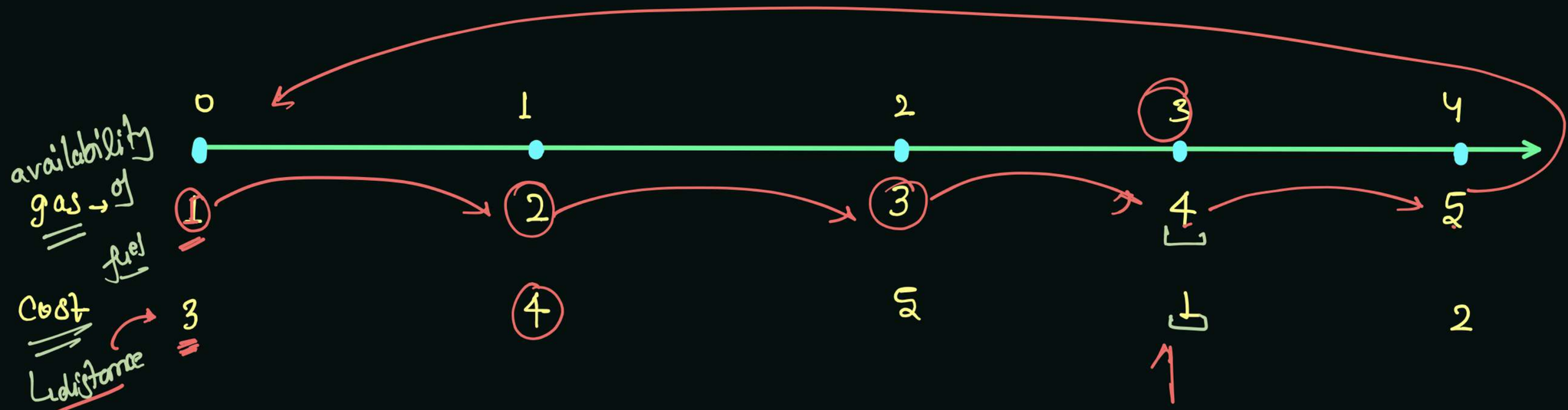
Gas Station

Friday, 29 October 2021 8:30 PM

0 1 2 3 4

gas = [1,2,3,4,5], cost = [3,4,5,1,2]

mileage - 1 km / 1 ltr



15 ltr
↳ total available

consumption
↳ 20 ltr.

infinite capacity of tank

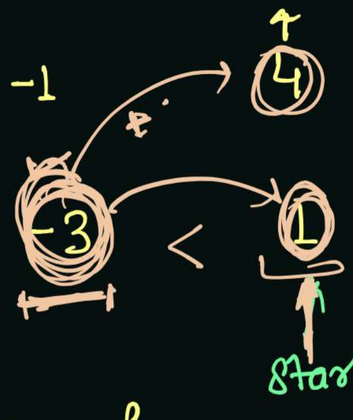
tank = ~~4 3 8 6 7 4 6 2 5~~ 0

× (1) Σ gas , & Σ cost

(*) Σ gas − Σ cost < 0 → Not possible

(1) Σ gas − Σ cost ≥ 0

gas →	4	1	3	10	8	7	2	3	2	6
cost →	4	3	4	6	11	1	5	4	3	1
gas - cost	0	-2	-1	4	-3	6	-3	-1	-1	4
prefix sum →	0	-2	-3	1	-2	4	1	0	-1	3



Result ⟶ Next Index from most -ve Element

Let's Analyse why it works ? i is most smallest

To prove - $gas[i+1] - cost[i+1] \geq 0$

$prefixSum[i+1] >= prefix\ sum[i]$

$prefixSum[i] + gas[i+1] - cost[i+1] \geq prefix\ sum[i]$

$gas[i+1] - cost[i+1] \geq 0$ — proved

Now from i+1 cum we move next gas station from i+2 ?

To prove → $gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] \geq 0$

$$prefixSum[i+2] \geq prefixSum[i]$$

$$prefixSum[i+1] + gas[i+2] - cost[i+2] \geq prefixSum[i]$$

$$\cancel{prefixSum[i]} + gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] \geq \cancel{prefixSum[i]}$$

$$\Rightarrow \underline{gas[i+1]} - \underline{cost[i+1]} + \underline{gas[i+2]} - cost[i+2] \geq 0$$

↳ proved

To prove →

$gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq 0$

$prefix[i+3] \geq prefixsum[i]$

$prefixsum[i+2] + gas[i+3] - cost[i+3] \geq prefixsum[i]$

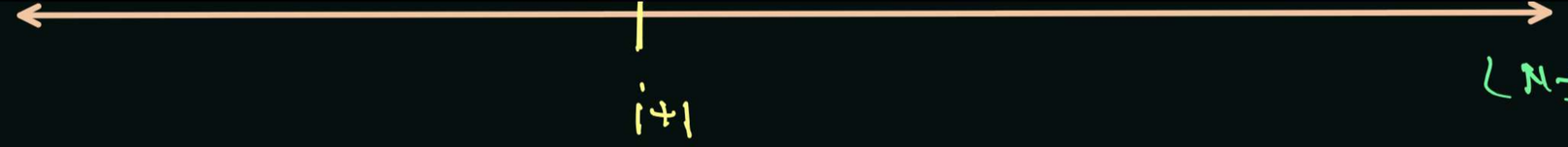
$prefixsum[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq prefixsum[i]$

~~$prefixsum[i]$~~ $+ gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3]$

$\geq$ ~~$prefixsum[i]$~~

$gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + gas[i+3] - cost[i+3] \geq 0$

proved



$$(gas[i+1] - cost[i+1]) + (gas[i+2] - cost[i+2]) + (gas[i+3] - cost[i+3]) + \cdots + (gas[N-1] - cost[N-1]) \geq 0$$

⟶ If we start from $i+1$, then we will reach at '0' index after jump from $(N-1)^{th}$ index

To prove → let see how we can reach from 0 to $i+1$ → +ve part

$\sum gas - \sum cost \geq 0$ ⟶ Because there is a result

$$gas[0] - cost[0] + \left[gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + \cdots + gas[N-1] - cost[N-1] \right]$$

$$gas[0] - cost[0] + gas[1] - cost[1] + \cdots + gas[i] - cost[i] + \left[gas[i+1] - cost[i+1] + gas[i+2] - cost[i+2] + \cdots + gas[N-1] - cost[N-1] \right]$$

Part 1 | Part 2

To prove ≥ 0

$$[gas[0] + gas[1] + gas[2] + \cdots + gas[N-1]] - [cost[0] + cost[1] + \cdots + cost[N-1]]$$

$$\sum gas - \sum cost \geq 0$$

$$\Downarrow$$

$$\Rightarrow \quad Part\ 1 + Part\ 2 \geq 0$$

$\Rightarrow$ we can from it $i+1$ in circular traversal.