Min Number of boats

Sunday, 12 September 2021 6:51 PM

weight of person will never cross the limit

boat ++

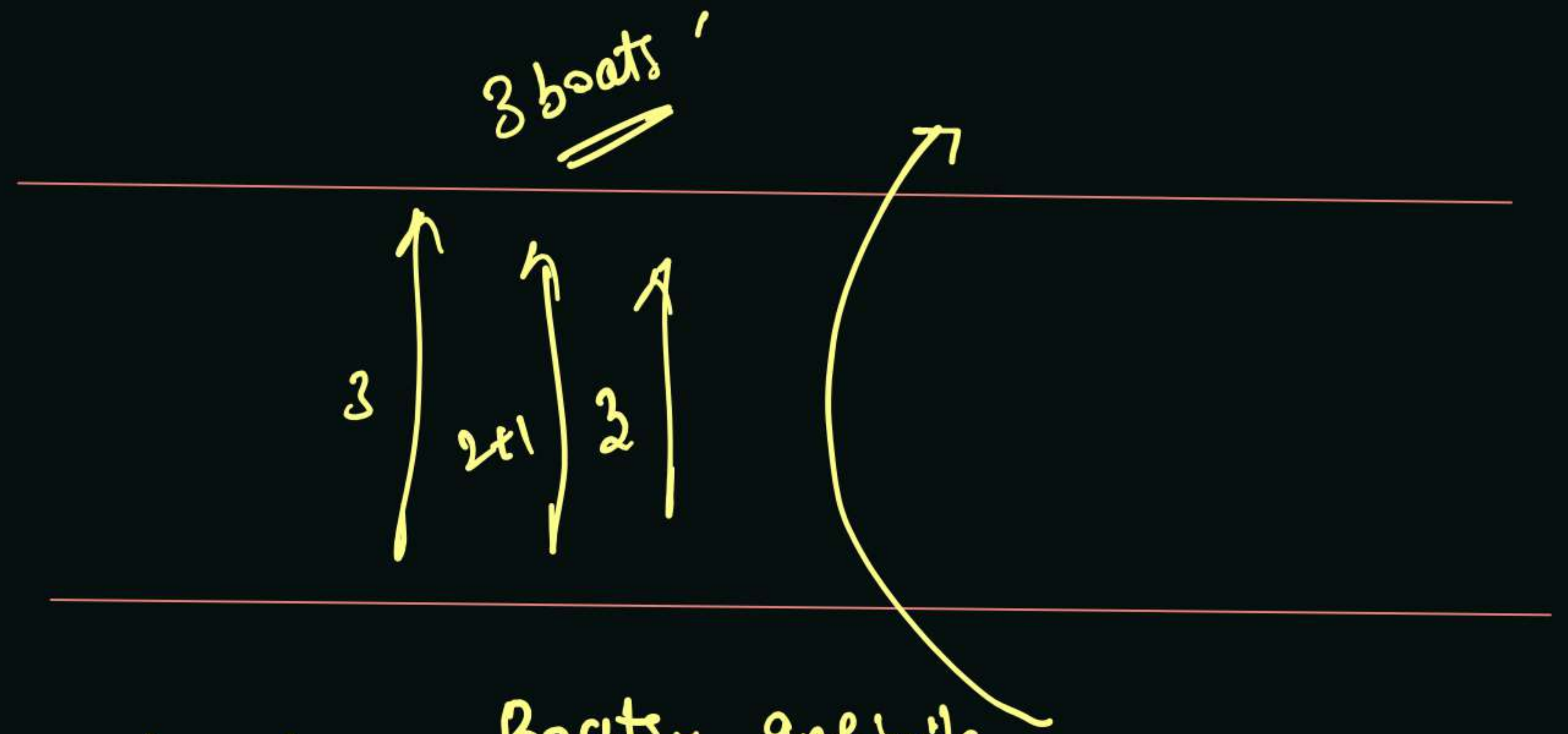
Cap = 10

sum = 11

7 2 3 4 7 6 2 1 4 5 6 10

sort

1 2 2 3 4 4 5 6 6 7 7 10

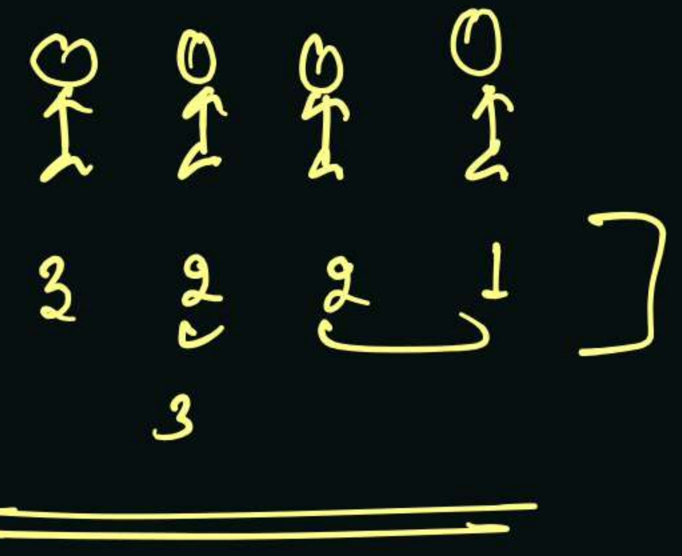


Boats - infinite

wt. limit

limit = 3

Max seat = 2



Min no. of boat = 3

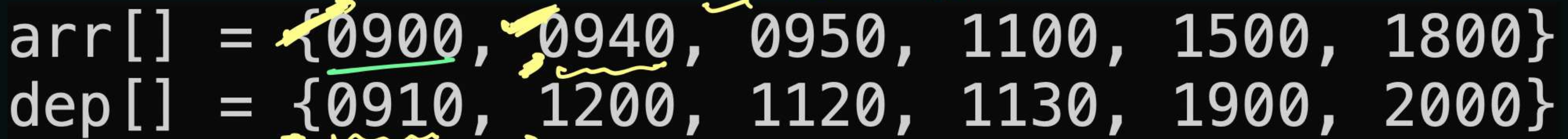
```
sum = arr[left] + arr[right]

if (sum <= cap){
    boat ++; left++; right--;
} else {
    boat ++; right--;
}
```

Min. Platforms

Sunday, 12 September 2021 7:28 PM

T1 T2 T3 T4 T5 T6



```
arr[] = {0900, 0940, 0950, 1100, 1500, 1800}
dep[] = {0910, 1200, 1120, 1130, 1900, 2000}
```

C-1

occupy platform

free platform.

Story→

Min Plateform Required for time table mongent of trains??

Actual task→

→ Max. no. of trains at a particular time is equal to no. of platforms required.

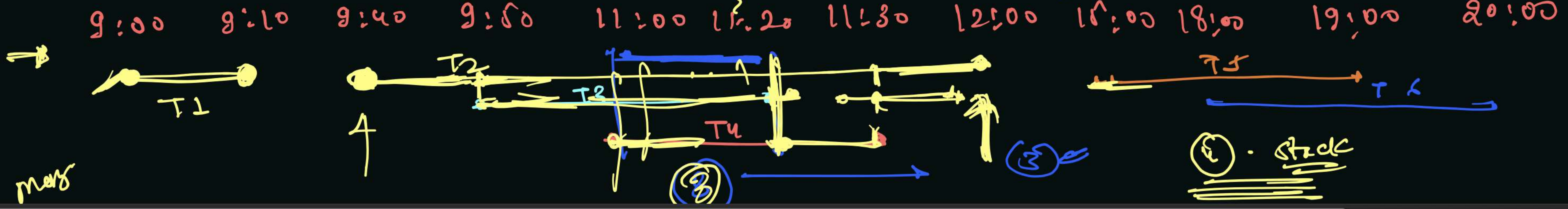
i → 9:00 i → 9:40 i → 9:50 i → 11:00 i → 15:00 i → 18:00

j → 9:10 j → 11:20 j → 11:30 j → 12:00 j → 19:00 20:00

cmax = ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ 2

omax = ~~0~~ ~~1~~ ~~2~~ 3

```
if( arr[i] <= dep[j]) {
    cmax++;    i++;
} else if( arr[i] > dep[j]){
    cmax--;  j++;
}
omax = Math.max (omax, cmax);
```



Print subarray with sum

Sunday, 12 September 2021 8:25 PM

Time - $O(n^3)$

$O(n^2)$ $O(n^3)$

array - [a b c d]

$O(n^2)$ $O(n^2)$ $O(n^2)$

Subarray

a → [a] → S1 → Sum of Subarr

ab → [ab] S2

abc → [abc] S3

abcd → S4

$O(n^2)$ → n

3.

b → S5

bc → S6

bcd → S7

c → S8

cd → S9

d → S10

Kadane's Algorithm

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

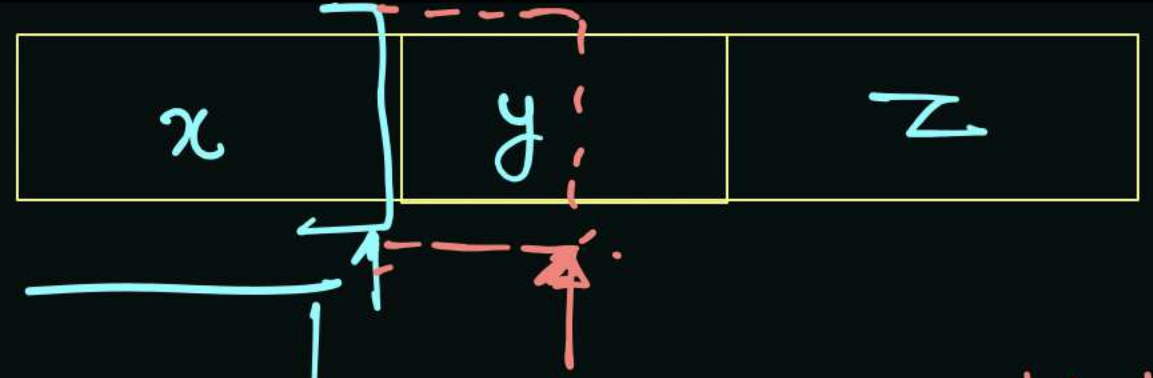
6

1

max sum subarr

Maximum Sum Subarray

All elements are +ve ⇒ take all Elements in a single subarra



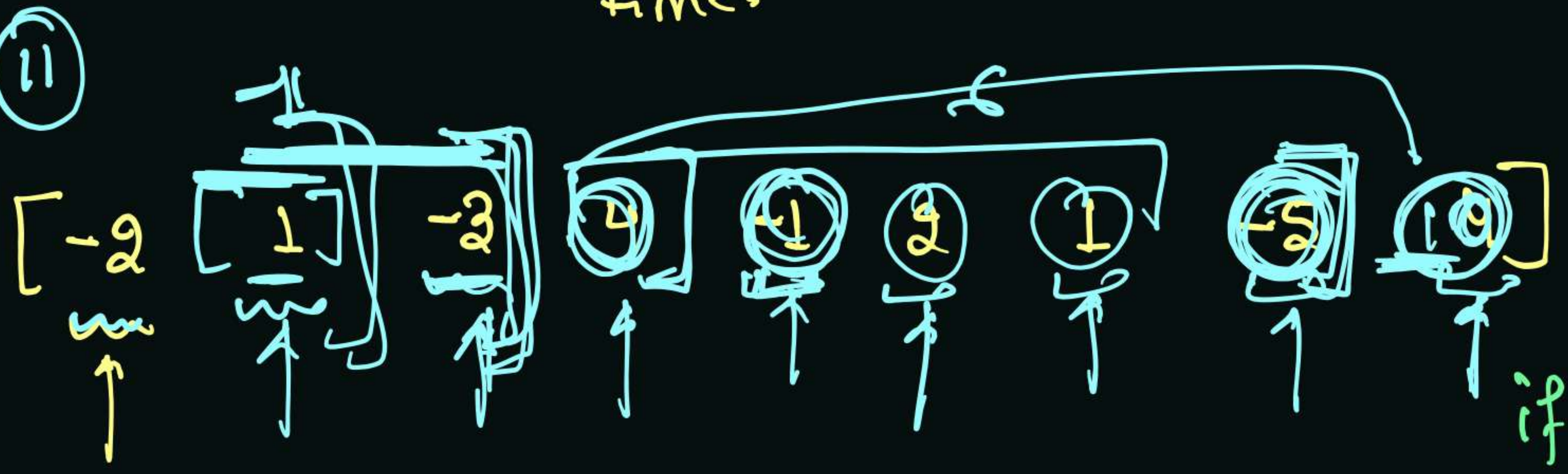
Sum of subarray → add in sum of subarray in previous part
→ start from itself.

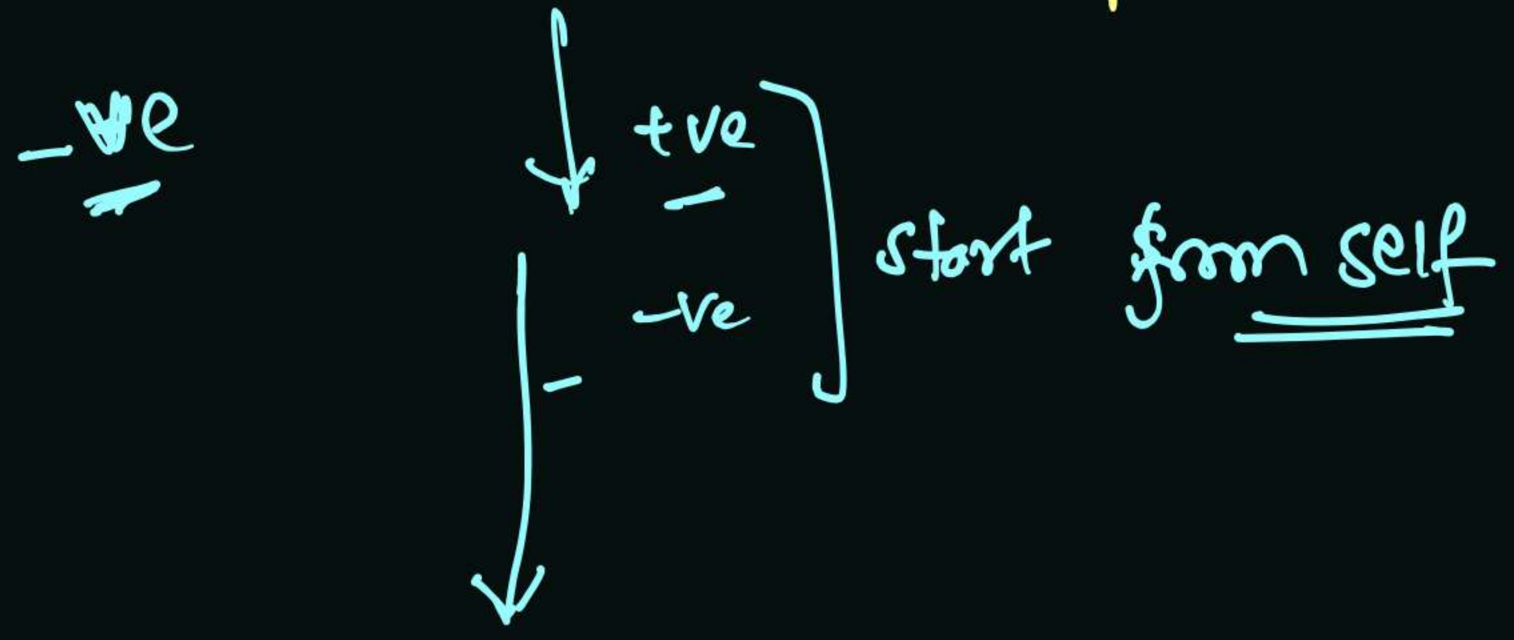
current sum, osum → max sum in a subarray

deal in Running time.

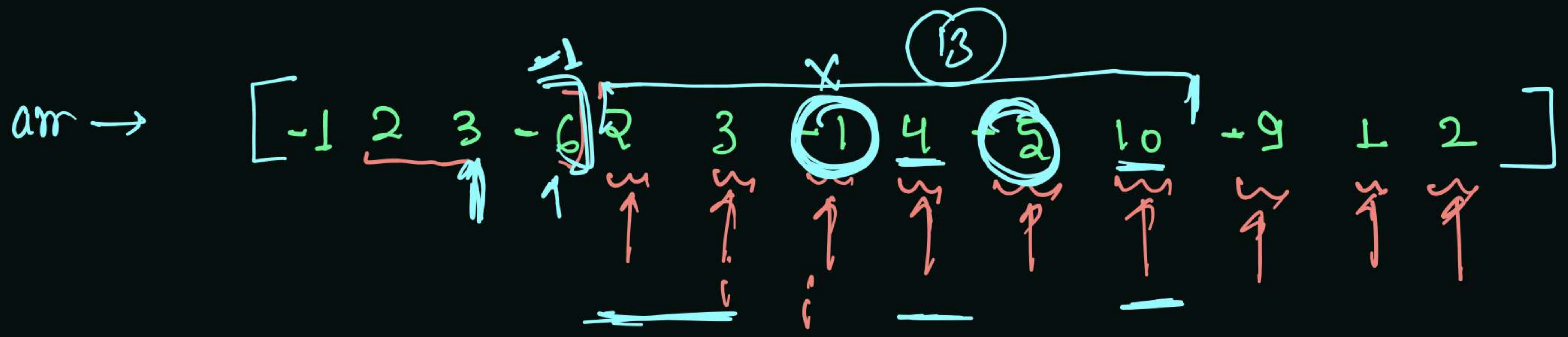
omax = ~~-∞~~ ~~-2~~ ~~1~~ ~~4~~ ~~5~~ ~~6~~ 11

csum = ~~0~~ ~~-2~~ ~~1~~ ~~-2~~ ~~4~~ ~~3~~ ~~5~~ ~~6~~ ~~1~~ 11





```
for(——){
    if (csum < 0) {
        csum = arr[i];
    } else {
        csum += arr[i];
    }
    omax = max(omax, csum)
}
```



csum = ~~0~~ ~~-1~~ ~~2~~ ~~5~~ ~~-1~~ ~~2~~ ~~5~~ ~~4~~ ~~8~~ ~~8~~ ~~13~~ ~~4~~ ~~5~~ 7

omax = ~~-∞~~ ~~-1~~ ~~2~~ ~~5~~ ~~9~~ 13

```
for( i =0 to i< leng) {
    if( csum < 0) {
        csum = arr[i];
        cstart= i;
    } else {
        csum += arr[i];
        cend = i;
    }
    omax = max (csum ~~+~~ vs omax)
}
```

Start

Increment

cstart

cend

ostart

oend

ostart to oend subarray

```
if(csum > osum) {
    osum = csum
    ostart = cstar;
    oend = cend;
}
```

K-Catenation

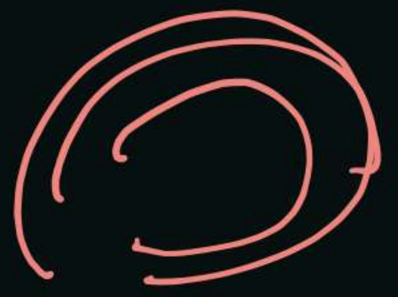
Sunday, 12 September 2021 9:32 PM

arr = {1, 2} , K = 3 Find max. sum subarray with K- concatinated array.

k times

virtual

$B = \underbrace{\{1, 2, 1, 2, 1, 2\}}_{k \text{ times arr}}$ → Max. sum subarray??



Constraints

① A size of array = 10^5

② range of K = 10^5.

⇒ we can't make an array with concatenation.

③ Time limit = 1 sec.

⇒ In 1 sec. we can do 10^9 operation.

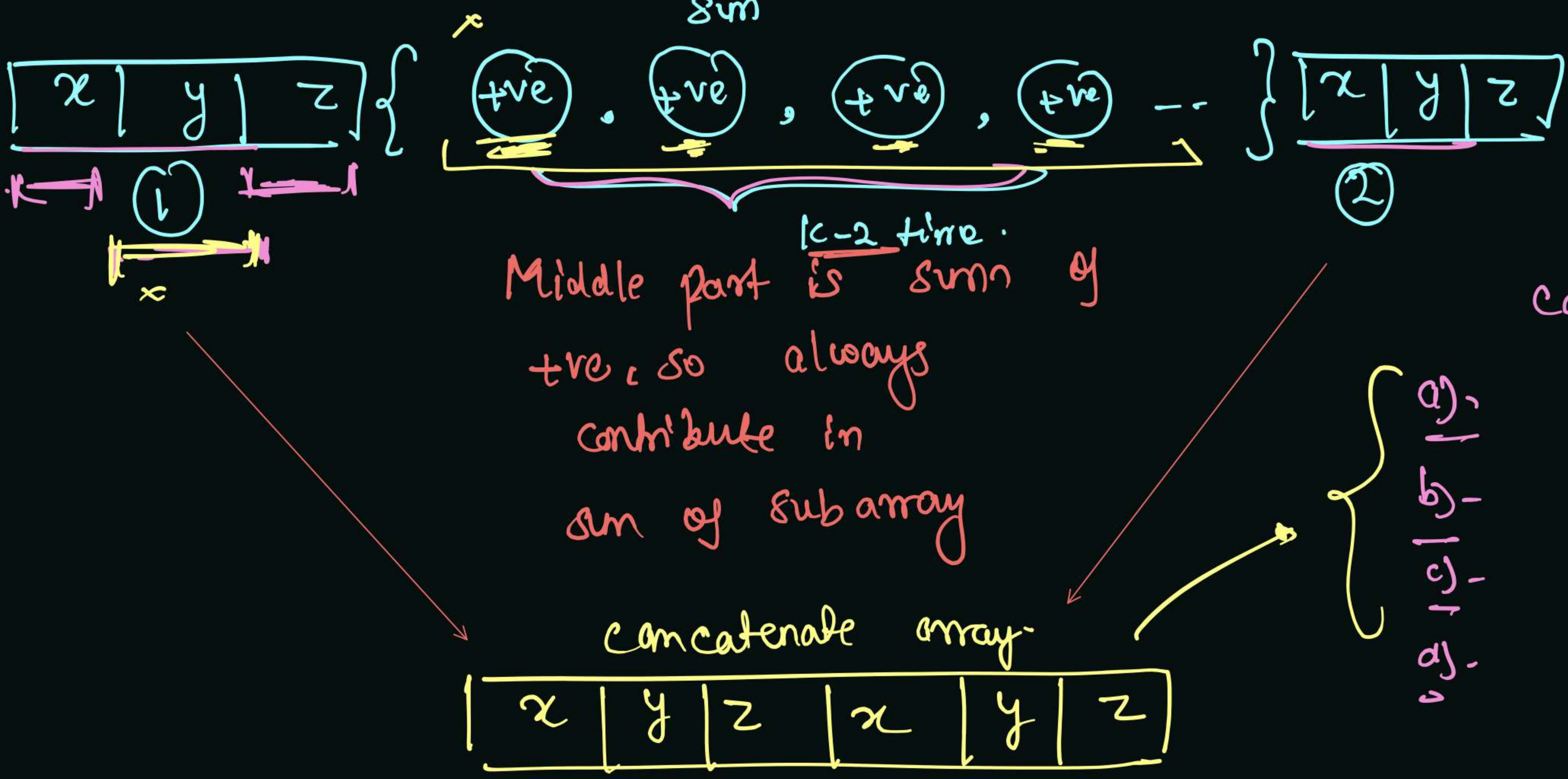
→ cyclic loop of K-concatenated array will not work.

Find Total sum of array ?? significance ?

Tsum

Case-I if(Tsum >= 0)

Tsum = +ve



Middle part is sum of +ve, so always contribute in sum of subarray

Sub case-s

a),
b)-
c)-
d).

Case-II. if(Tsum < 0)

Kadane's -

20 -160 20 20 -160

Tsum = -80