1043. Partition Array for maximum Sum

arr = [1, 15, 7, 9, 2, 5, 10] , K=3

ans = 84

explaintion

arr = [1, 15 | 7, 9, 2 | 5, 10]   K=3

      15 15 | 9 9 9 | 10 10

arr = [1, 15, 7 | 9 | 2, 5, 10]

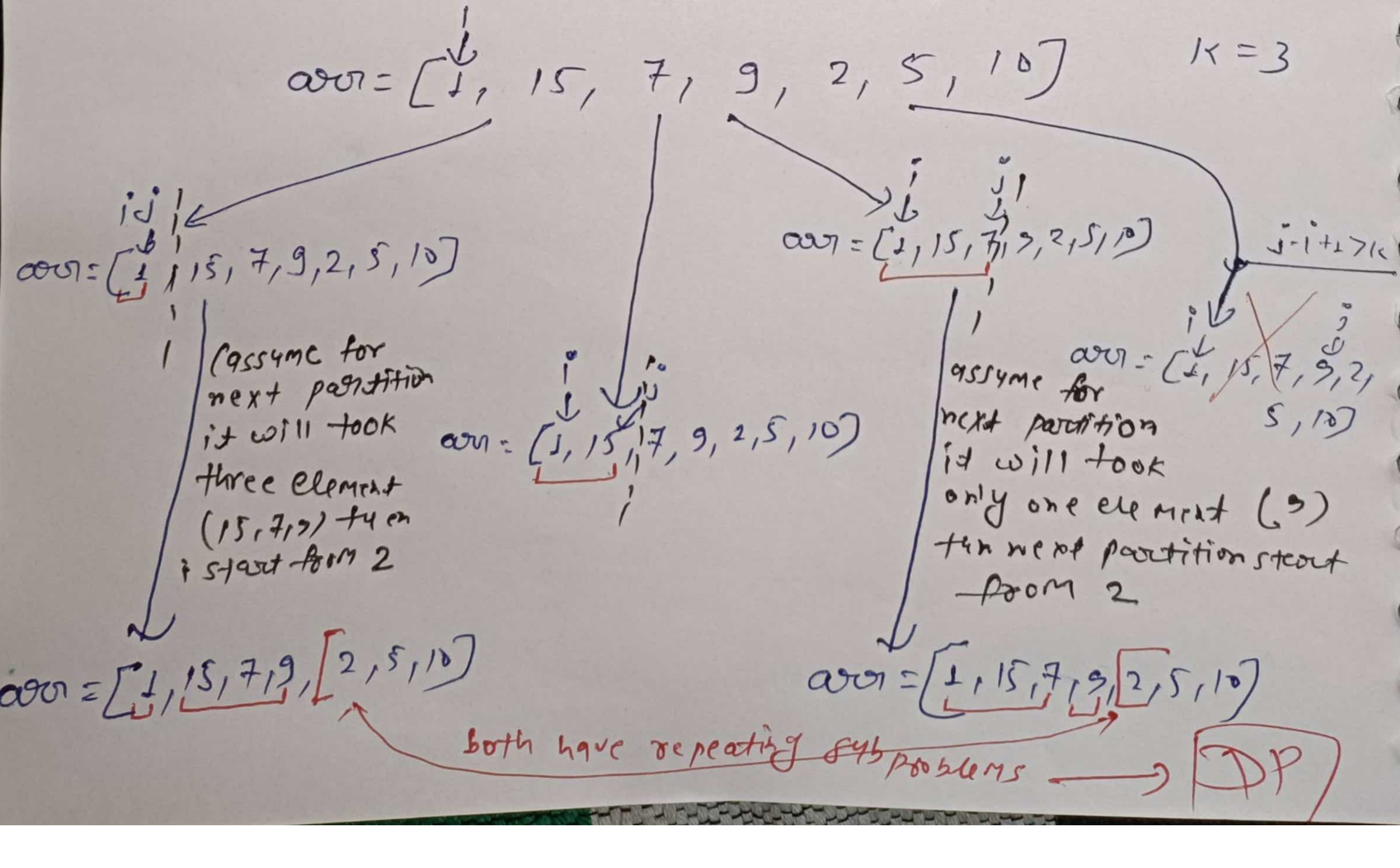
      15 15 15 | 9 | 10 10 10

soln

we try all possibility to place the partition

& stars + Bars [

↳ T.C. = exponential / factorial



i
arr = [1, 15, 7, 9, 2, 5, 10]
K=3
i j
arr = [1, 15, 7, 9, 2, 5, 10]
(assume for next partition it will took three element (15, 7, 9) then i start from 2
arr = [1, 15, 7, 9, [2, 5, 10]
i j
arr = [1, 15, 7, 9, 2, 5, 10]
i j
arr = [1, 15, 7, 9, 2, 5, 10]
assume for next partition it will took only one element (9) then next partition start from 2
arr = [1, 15, 7, 9, [2, 5, 10]
j-i+1>K
i j
arr = [1, 15, 7, 9, 2, 5, 10]
both have repeating sub problems → DP

Imp

Recursion:-

j = i; j < min(n, i+k) ; j++

why we wrote this

sub array size can be ~~+~~ at most K so j < i+K

if our i is at

[1, 15, 7, 9, 2, 5, 10]
↑ i

i+K = 6+3 = 9.
but n = ~~7~~

so j<n

T.C. = exponential / factorial

~~memoization~~

solve (i, arr) → This will return the largest sum of the given array after partition

or.

[1, 15, [7, 9, 2, 5, 10] , k = 3

solve (2, arr) → from 2nd index to n the largest sum after partition this fn will return

memoization

dp → 1D array

dp[i] = solve (i, arr);

Bottom-UP

Top-Down
start idx = 0 & go till (n-1)

in Bottom-UP
start idx = n-1 & go till 0

$$\boxed{\begin{aligned} T.C. &= O(n*k) \\ S.C. &= O(n) \end{aligned}}$$

Space Optimization

we can optimize space from O(n) to O(k)

use dp[k+1] ← initially all values = 0.

Imp

$dp[i \% k] = max(dp[i], (maxVal * (j-i+1) + dp[(j+1) \% k]));$

↳ this will give you wrong ans bcz.

$6 \% 2 = 0$ & $8 \% 2 = 0$

↳ for two different values % can gives same answer.

In top-Down approch

$$dp[i] = \max(dp[i], (maxVal * (j-i+1)) + dp[j+1]))$$

These both are different but in ~~MOD~~ %. while space optimization these can be same at some point.

- So we use an ans variable