K = 2

1 1 0 0 0 1 1 1

sp, ep

ZC = 2
max = 5

1. initialize
2. Stop till when while(ZC > K)
3. calc ans
4. expand

mod 1 => A -> 2010, 2012, 2022

mod 3 =>

10

10 5 2 5

sp, ep, k = 100

5 2 5
2 5
5

Window prod = 50

Ans => 1 + 2 + 2 + 3

1. initialize with new person
2. Shrink window until it's valid
3. Calculate ans
4. expand window

expand => new person gets * by product window

shrink => person who is leaving the window divide the window by its value

window

15

10

2 2 3 {3 3 4 4 5 6 6}

i, j, l, r

l++

Prev => 4

{2 2 5 6}

{2 3 4 6} => {2 3 4 6}

Move any pointer until it is not staying on the same number

Two Sum

10

1 3 4 6 7 8

0 1 2 3 4 5

9 < 10 [1,4]

11 > 10 [2,5]

for(i){

i, j, j+1, 2 sum, n-1

for(j){

targ - (a[i] + a[j])

2 sum

}

}

2 4 8 1 1

a[l] + a[r] = target - (a[i] + a[j])

Four Sum

a b c d => target

1) Four Indexes which are summing up to target
2) Indexes should be distinct
3) {2 1 2 4} {2 4 1 2} Distinct quadruples

0 1 2 3 0 3 1 2