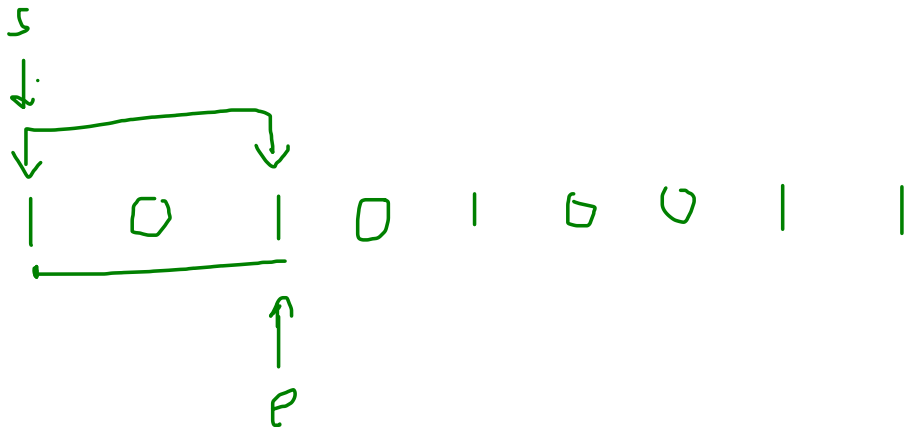
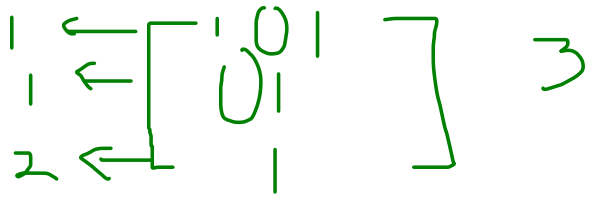
goal 2



count num of subArr with sum = goal



Window size

Sum of subArr <= goal --> goal + (goal-1) + (goal - 2) ...

Sum of SubArr = 2, 1, 0 ...

Sum of SubArr <= (goal - 1) ==> 2 - 1 = 1

1, 0, ...

goal = 5

(1). sum of all subArr <= goal --> 5, ~~4, 3, 2 , 1, 0~~

(2). sum of all subArr <= goal - 1 --> 5 - 1 = ~~4, 3, 2, 1, 0~~

final subArr count with sum = goal ==> (1) - (2)

why not recursion , as f(x) is not function of x it is some constant

f(x) = f(constant) - f(constant-1)

count of all subArr with sum <= target value

goal = 2

ans = (window len) = ei - si + 1

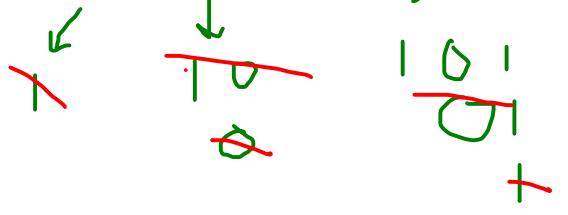
sumWindow > targetValue

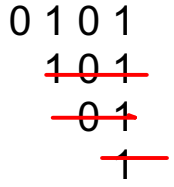
1. tar = goal
2. tar = goal - 1

final ans = 1 - 2

sumWindow = 0 1 + 0 = 1 2 3 2 3 2

ans = 0 = 1 + 2 + 3 + 4 + 4 + 5 + 6 + 5 + 4





= 34

aquire --> everyTime

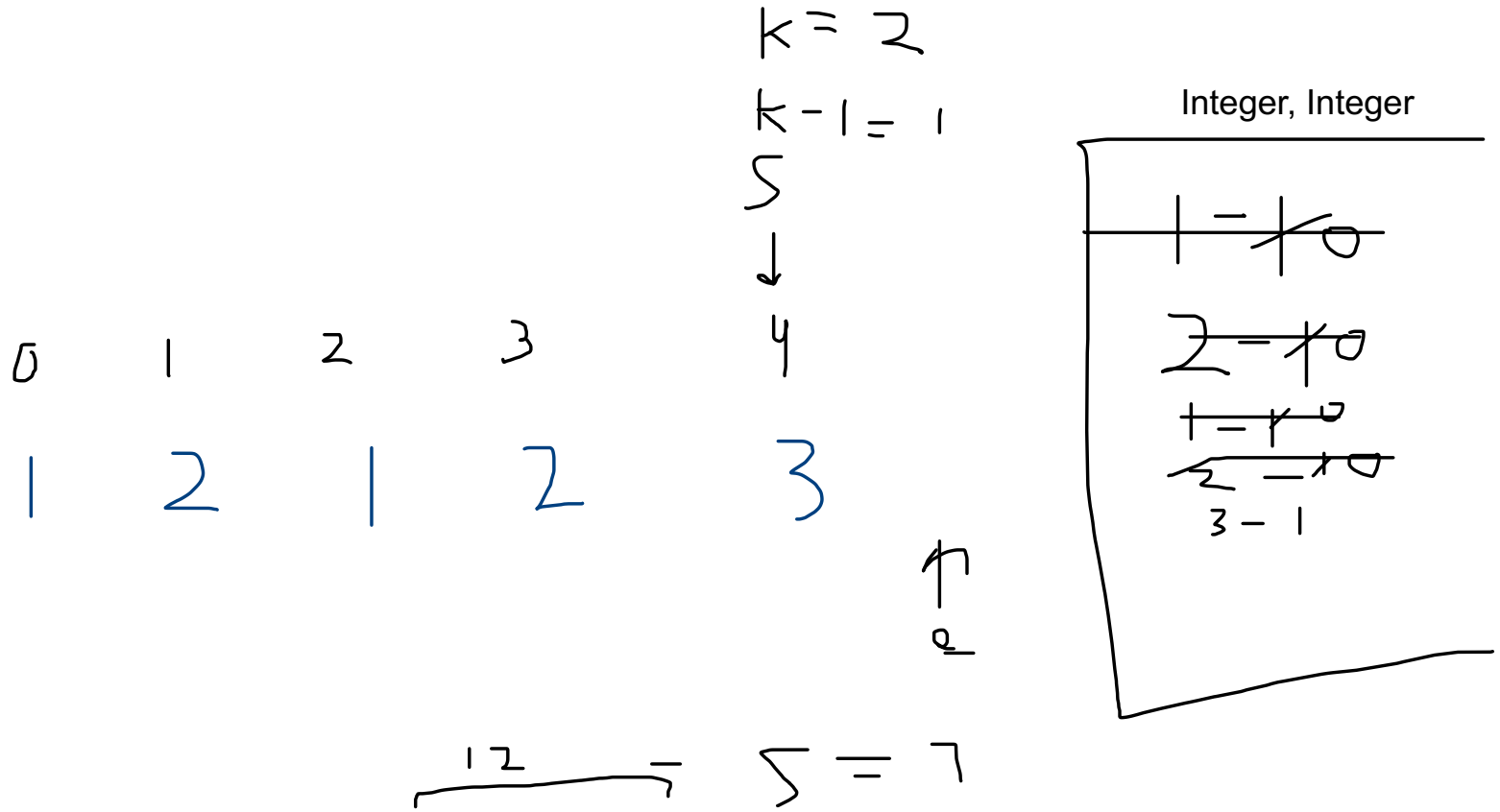
Release --> while(windowSum > targetValue)

record Ans --> everyTime

k 2

2 2 1 1 2 2 1 1 2

count subArr with exactly k odd integer = (count subArr with <= k) - (count subArr with <= k - 1)



count subArr with exactly k distinct integer = (count subArr <= k) - (count subArr <= k - 1)

ans = 0 + 1 + 1 + 1 + 1 + 1 = 5

distinct Integer in window = ~~0 + 2 + 2 + 2 + 2~~