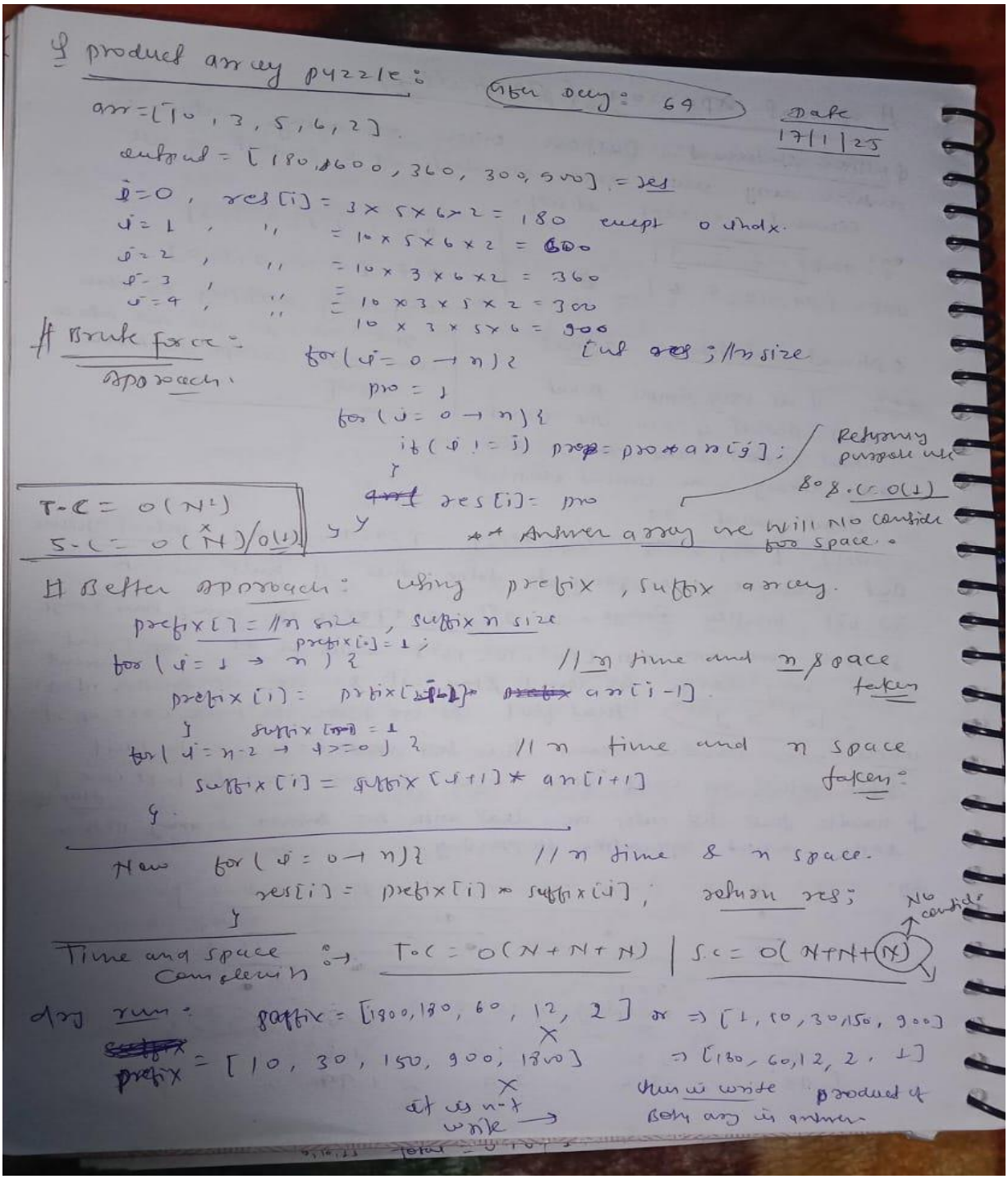
*1st PAGE : Question Problem :* ***Product Array PUZZLE***

## Product of Array Except Self

PREFIX , ARRAY Problem : Leetcode (238)



# Optimal Approach: (Practically)

Question statement: Question given an array, return an answer array where each element is a product of all element except at self.

eg: arr=[1, 2, 3, 4]

ans = [24, 12, 8, 6] _ Ans

arr = [-1, 1, 0, -3, 3].

ans = [0, 0, 9, 0, 0].

any thing multiply by zero give zero so all the place have zero except that one itself:

Optimal solution: Ideal

*** it is very simple what find product of all the elemet total product: and store in this array dirctly with current element.

total product = 24.

ans[] = [24/1, 24/2, 24/3, 24/4] = [24, 12, 8, 6] Actual Answer

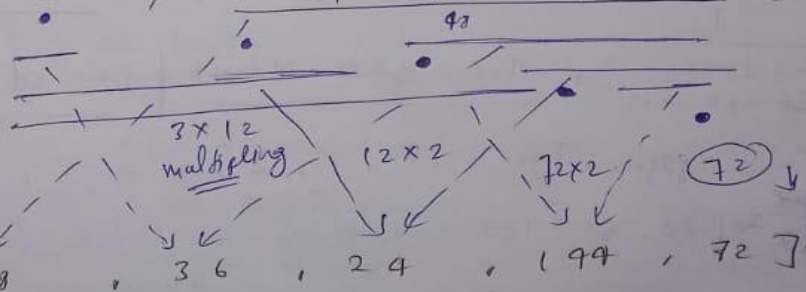
But when we are going to solve this it fails due to 32 bit overflow range, $2^{32}$ = 2147483647 cross this range

Suppose we have arr=[$10^9$, $10^7$, $10^5$] what is Ans Total product very large we can't store it so we use another idea. $10^2 > 2^{32}$ that fail so we have to take care of it.

when zero involve then, this test case also fail total result we can't find here so we have to take care of this also

to handle these test cases we deal with the answer & array before how what operation happening.

eg: [ 1 , 4 , 6 , 1 , 2 ]

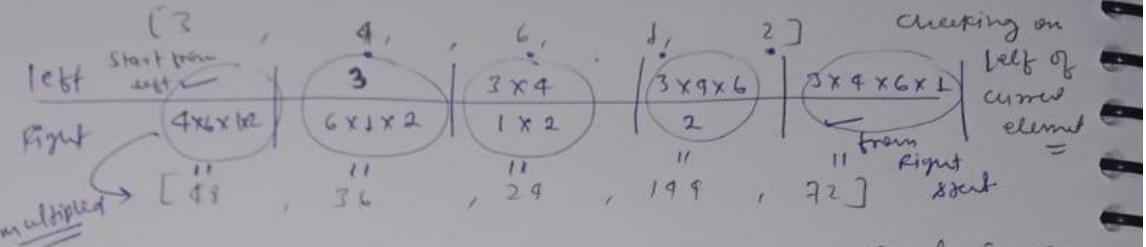


# So what we are doing over here :-

result [i] = Multiply all on left * multiply all on Right

** we can handle here those both cases:

→ So we need to maintain two product one is from the left to that point and second is from Right to that point




T.C: O(3N)
T.C: O(2N)
O(N) if return ans not any

This is Better Approach: practical Brute force Approach: when we use two array for store left & Right then it took O(N) space for computing Quickly task.

X

But we can optimized it using prefix & suffix variable. in, then (2N) prefix, suffix space will optimized to constant.

→ Initially take a ans array: input = [3, 4, 6, 1, 2]

→ update it using single loop calculating prefix product

ans[0] = 1

ans = [1, 3, 12, 72, 72 ]

→ Now why suffix variable and input array we have to store result iterably input ans from last:

suffix = 1

ans = [48, 36, 24, 144, 72] ← ans[i] = ans[i] * suffix.

suffix = suffix * arr[i]

& 2x1 = 2
1x2 = 2
2x6 = 12, 12x4 = 48

result = O(2N) for space = O(1) only O(N) for result returning