

## Product of Array Except Self

PREFIX , ARRAY Problem : Leetcode (238)

Product array puzzle:

arr = [10, 3, 5, 6, 2]

output = [180, 600, 360, 300, 900] = res

i = 0, res[i] = 3 x 5 x 6 x 2 = 180 except 0 index.  
 i = 1, " = 10 x 5 x 6 x 2 = 600  
 i = 2, " = 10 x 3 x 6 x 2 = 360  
 i = 3, " = 10 x 3 x 5 x 2 = 300  
 i = 4, " = 10 x 3 x 5 x 6 = 900

Brute force:

for (i = 0 to n) {  
 prod = 1  
 for (j = 0 to n) {  
 if (j != i) prod = prod \* arr[j];  
 }  
 res[i] = prod  
 }

T.C =  $O(N^2)$   
 S.C =  $O(N)$  /  $O(1)$

Better approach: using prefix, suffix array.

prefix[] = // n size, suffix[] = n size  
 prefix[0] = 1;  
 for (i = 1 to n) {  
 prefix[i] = prefix[i-1] \* arr[i-1];  
 }  
 suffix[n] = 1;  
 for (i = n-1 to 0) {  
 suffix[i] = suffix[i+1] \* arr[i+1];  
 }

Now for (i = 0 to n) {  
 res[i] = prefix[i] \* suffix[i];  
 }

Time and space complexity:  $T.C = O(N + N + N)$  |  $S.C = O(N + N + N)$

dry run: prefix = [1, 3, 15, 90, 180] or [1, 10, 30, 150, 900]  
 suffix = [180, 60, 12, 2, 1]  
 at each index work →

thus we write product of both array in answer

17/1/25

## # Optimal Approach: (Practically)

Question Statement: Question gives an array, return an answer array where each element is a product of all elements except itself.

eg:  $arr = [1, 2, 3, 4]$   
 $ans = [24, 12, 8, 6]$  - ~~is~~

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

$ans = [0, 0, 0, 0, 0]$

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 element  
 total product: and store in this  
 array dividing with current element.

total product = 24

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

But when we are going to solve this it fails due to

32 bit overflow range,  $2^{32} = 2147483648$  covers this range

Suppose we have  $arr = [10^9, 10^7, 10^5]$  what is ~~is~~ Total product  
 very large we can't store it so we use another idea.

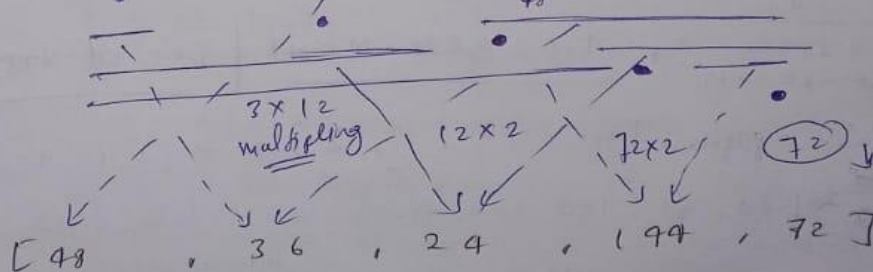
$10^2 > 2^{32}$  that fail so we have to take care of it.

when zero involve then, this test case also fail

total result we can't find here so we have to take care of this

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

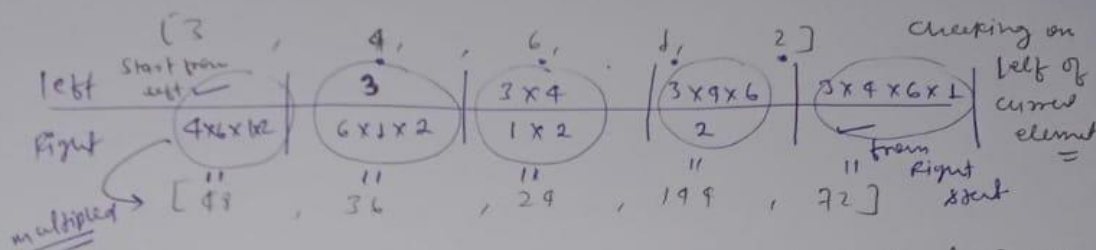
eg:  $[3, 4, 6, 1, 2]$



# So what we are doing over here:-

result[i] = multiply all on left + multiply all on right  
we can handle the make some cases:

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



This is Better Approach: product brute force approach:  
 when we use two array for store left & right then  
 it took  $O(N)$  space for computing every task

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 using suffix variable and input array  
 we have to store result iteratively input array  
 from last:

suffix = 1  
 ans = [98, 36, 24, 199, 72] ← ans[i] = ans[i] \* suffix

space =  $O(2N)$  for  
 space =  $O(1)$   
 only  $O(N)$  for result  
 return

suffix = suffix \* arr[i]  
 &  
 2 \* 1 = 2  
 1 \* 2 = 2  
 2 \* 6 = 12, 12 \* 4 = 48