Question: https://leetcode.com/problems/product-of-array-except-self/

Constraints: You must write an algorithm that runs in O(n) time and without using the division operation.

Lets think of it simply, if we could have used division operator and array elements were non zero we could have simply done res[i] = totalProduct/ar[i]

But since neither can we use division nor are the elements non-zero so we can use a solution similar to prefix sum approach.

Instead of sum we calculate prefix product.

Now if we iterate from 0 to nth pos and maintain prefix product, and then re iterate from nth pos to 0 and maintain another prefix product, and multiply both the results that will give us the answer.

Code:  
class Solution {

public int[] productExceptSelf(int[] nums) {

int l = nums.length, res[] = new int[l];

int p = 1;

//left prefix

for(int i=0; i<l; i++){

res[i]=p;

p\*=nums[i];

}

//right prefix

p=1;

for(int j=l-1; j>=0; j--){

res[j]\*=p;//multiplying right subarray product and left subarray product

p\*=nums[j];

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>