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

**Product of Array Except Self**

**Given an integer array nums, return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i].**

**The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.**

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

Example 1:

Input: nums = [1,2,3,4]

Output: [24,12,8,6]

Example 2:

Input: nums = [-1,1,0,-3,3]

Output: [0,0,9,0,0]

Constraints:

2 <= nums.length <= 105

-30 <= nums[i] <= 30

The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

**Method 1: (Brute Force)**

For each element multiply all other elements.

Time Complexity: O(n2) *[two nested loops]*

Space Complexity: O(1)

**Method 2: (prefix product and postfix product)**

Store prefix product of each element in one array

Store postfix product of each element in another array

Multiply the two arrays to obtain final answer.

[can be done without using prefix array and postfix array in O(1) space complexity using answer array]

Time Complexity: O(n) *[sorting]*

Space Complexity: O(1) *[output array doesn’t count as extra space complexity]*

vector<int> productExceptSelf(vector<int>& nums) {

        vector<int> answer;

        int prev=1;

        for(int i=0; i<nums.size(); i++){

           answer.push\_back(prev);

           prev\*=nums[i];

        }

        prev=1;

        for(int i=nums.size()-1; i>=0; i--){

            answer[i]\*=prev;

            prev\*=nums[i];

        }

        return answer;

    }