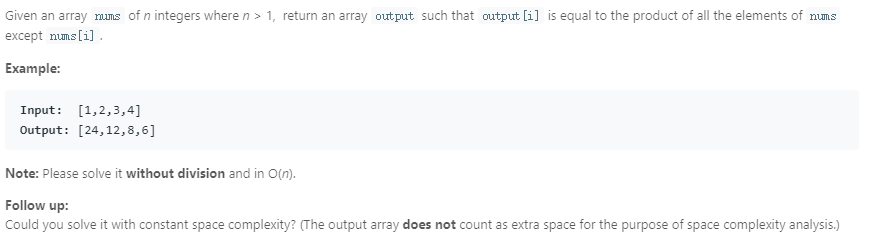
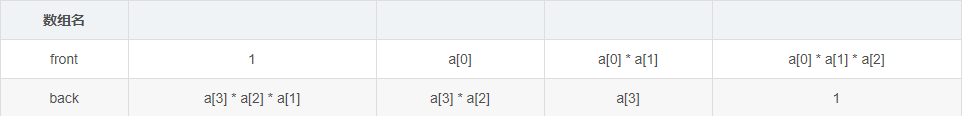
# 238. Product of Array Except Self (Medium)



思路：

数组a包含4个元素。

构建如下两个数组front和back：



通过将这两个数组的元素对应相乘，就能得到题目要求的，除了自身以外所有数组元素的乘积。

第一个数组front的构建方式是temp设置为1，从第2位开始，使用一个变量tmpResult存储从a[0]到当前a[i -1]的乘积，每次都把这个乘积赋给front[i]。

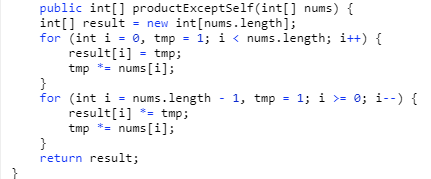
第二个数组back的构建方式类似，不过从数组a最后一个元素开始，设置back[n - 1]为1(n 为数组a的大小)，从倒数第二位开始，使用一个变量tmpResult存储从a[n - 1]到当前a[i + 1]的乘积，每次都把这个乘积赋给back[i]。

最后将这两个数组元素对应相乘即得到结果。

为了节省空间，可以直接使用要返回的数组result作为那两个数组：首先是作为front数组，操作过程一样，然后作为back数组时是直接在当前元素上乘以temProduct。

为了构建这两个数组需要两个for循环，它们的区别是遍历数组a的顺序不同，所以还可以合并，只用一个for循环同时进行对product[i] 和 product[n - i -1]的操作。

Java:



从最左开始，当从第一个开始的时候先设个tmp=1去替代自己，然后剩余求积。

从最右开始，但是这个时候最左的流程已经结束了，所以要倒序，而倒序的时候result[]是从自己开始的，所以起始乘的也是1.

C++:



