

给定一个整型数组，在数组中找出由三个数组成的最大乘积，并输出这个乘积。

示例 1:

输入: [1,2,3]

输出: 6

示例 2:

输入: [1,2,3,4]

输出: 24

注意:

1. 给定的整型数组长度范围是[3,104]，数组中所有的元素范围是[-1000, 1000]。
2. 输入的数组中任意三个数的乘积不会超出32位有符号整数的范围。

思路：直接sort排序，比较第1，2位数乘上最后一位和最后三位数相乘的大小即可，情况可以分为1个正整数2个正整数和3个正整数，在情况1、2中，最大的负乘以最大的正，结果最大，第三个需要比较下大小。

```
class Solution {
public:
    int maximumProduct(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        int n=nums.size()-1;
        return nums[0]*nums[1]*nums[n]>nums[n]*nums[n-1]*nums[n-2]?
nums[0]*nums[1]*nums[n]:nums[n]*nums[n-1]*nums[n-2];
    }
};
```