/\*

一看就是动态规划

解题思路：乘法与加法最大差别在于，当前元素的符号具有全局性的作用。

如果当前元素为负，那么连乘到上个元素的最大乘积，再乘以当前元素，就变成负数，甚至可能成为最小乘积。

同样，连乘到上个元素的最小乘积如为负，再乘以当前元素，就变成正数，甚至可能成为最大乘积。

因此使用动态规划的方法：

记maxLast/minLast为连乘到上个元素的最大/小乘积

记maxCur/minCur为连乘到当前元素的最大/小乘积

记maxAll为全局最大乘积

\*/

class Solution {

public:

int maxProduct(vector<int>& nums)

{

if(nums.size()==0)

return 0;

if(nums.size()==1)

return nums[0];

int all=nums[0];

int lastmax=nums[0];

int lastmin=nums[0];

int curmax=nums[0];

int curmin=nums[0];

for(int i=1;i<nums.size();i++)

{

curmax=max(nums[i],max(lastmax\*nums[i],lastmin\*nums[i]));

curmin=min(nums[i],min(lastmax\*nums[i],lastmin\*nums[i]));

lastmax=curmax;

lastmin=curmin;

all=max(all,curmax);

}

return all;

}

};