

## 三个数的最大乘积

给你一个整型数组 `nums`，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: `nums = [1,2,3]`

输出: 6

示例 2:

输入: `nums = [1,2,3,4]`

输出: 24

示例 3:

输入: `nums = [-1,-2,-3]`

输出: -6

```
int cmp(const void* a,const void* b)
{
    return *(int*)a-*(int*)b;
}
/*
```

如果数组中全是非负数，则排序后最大的三个数相乘即为最大乘积；如果全是非正数，则最大的三个数相乘同样也为最大乘积。

如果数组中有正数有负数，则最大乘积既可能是三个最大正数的乘积，也可能是两个最小负数（即绝对值最大）与最大正数的乘积。

综上，我们在给数组排序后，分别求出三个最大正数的乘积，以及两个最小负数与最大正数的乘积，二者之间的最大值即为所求答案。

```
*/
int maximumProduct(int* nums, int numsSize){

    qsort(nums,numsSize,sizeof(int),cmp);
    int s1=nums[numsSize-1]*nums[numsSize-2]*nums[numsSize-3];
    int s2=nums[0]*nums[1]*nums[numsSize-1];
    return s1>s2?s1:s2;
}
```