

有序数组的平方

给你一个按 **非递减顺序** 排序的整数数组 `nums`，返回 **每个数字的平方** 组成的新数组，要求也按 **非递减顺序** 排序。

示例 1:

输入: `nums = [-4,-1,0,3,10]`

输出: `[0,1,9,16,100]`

解释: 平方后, 数组变为 `[16,1,0,9,100]`

排序后, 数组变为 `[0,1,9,16,100]`

示例 2:

输入: `nums = [-7,-3,2,3,11]`

输出: `[4,9,9,49,121]`

```
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
int cmp(const void* a,const void* b)
{
    return (*(int*)a-*(int*)b);
}
int* sortedSquares(int* nums, int numsSize, int* returnSize){
    //原数组平方后存储
    for(int i=0;i<numsSize;i++)
    {
        nums[i]=nums[i]*nums[i];
    }
    qsort(nums,numsSize,sizeof(int),cmp);
    *returnSize=numsSize;
    return nums;
}
```