

## K 次取反后最大化的数组和

给定一个整数数组  $A$ ，我们**只能**用以下方法修改该数组：我们选择某个索引  $i$  并将  $A[i]$  替换为  $-A[i]$ ，然后总共重复这个过程  $K$  次。（我们可以多次选择同一个索引  $i$ 。）

以这种方式修改数组后，返回数组可能的最大和。

示例 1:

输入:  $A = [4, 2, 3]$ ,  $K = 1$

输出: 5

解释: 选择索引 (1,)，然后  $A$  变为  $[4, -2, 3]$ 。

示例 2:

输入:  $A = [3, -1, 0, 2]$ ,  $K = 3$

输出: 6

解释: 选择索引 (1, 2, 2)，然后  $A$  变为  $[3, 1, 0, 2]$ 。

示例 3:

输入:  $A = [2, -3, -1, 5, -4]$ ,  $K = 2$

输出: 13

解释: 选择索引 (1, 4)，然后  $A$  变为  $[2, 3, -1, 5, 4]$ 。

```
bool cmp(int a,int b)
{
    return abs(a)>abs(b);
}
class Solution {
public:
    int largestSumAfterKNegations(vector<int>& A, int K) {
        sort(A.begin(),A.end(),cmp);
        int res=0;
        for(int i=0;i<A.size();i++)
        {
            if(A[i]<0&&K>0)
            {
                A[i]=-A[i];
                K--;
            }
        }
        for(int i=0;i<A.size();i++)
            res+=A[i];
        return res;
    }
};
```

```
        }
        res+=A[i];
    }
    K=K%2;
    res-=K*2*A[A.size()-1];//偶数减去 0 奇数减去最后一个（最小）的 2 倍
    return res;
}
};
```