

## 1.题目名称

魔法优惠券

## 2.代码行数

47行

## 3.算法思想

先从小到大排序，从小到大遍历。

再从大到小遍历。

## 4.主要/核心函数分析

FindMax

```
1  int FindMax(vector<ll> value ,vector<ll> cons,int n,int m){           //寻找最大
    回报
2      sort(value.begin(),value.end());
3      sort(cons.begin(),cons.end());
4      ll sum=0;
5      for(int i=0;i<min(n,m);i++){           //从小往大遍历
6          if(value[i]<0 && cons[i]<0)        sum+=value[i]*cons[i];
7      }
8      reverse(value.begin(), value.end());
9      reverse(cons.begin(),cons.end());
10     for(int i=0;i<min(n,m);i++){           //从大往小遍历
11         if(value[i]>0 && cons[i]>0)        sum+=value[i]*cons[i];
12     }
13     return sum;
14 }
```

先从小到大排序，从小到大遍历,优惠券及价值均为负则相乘。

再倒序，从前往后遍历，优惠券及价值均为正则相乘。

## 5.测试数据(规模,测试次数)

规模:N和M在 $[1,10^6]$ 之间，所有的数据大小不超过 $2^{30}$ 。

测试次数:3

测试用例:见测试文件

## 6.运行结果

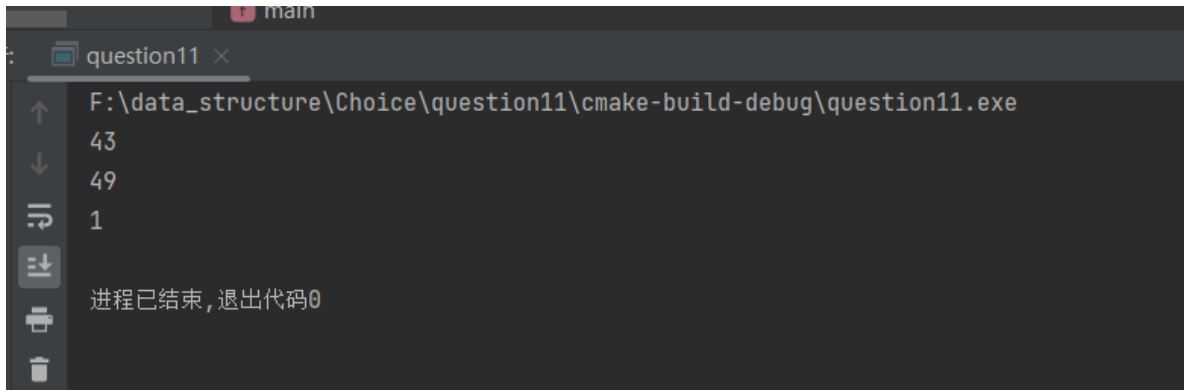
第i行对应第i个测试用例。

```
1  43
2  49
3  1
```

## 7.时间复杂度分析

时间复杂度最大的是sort排序，因此时间复杂度为 $O(n\log n)$ 。

## 8.结果截屏图片



```
question11 x
F:\data_structure\Choice\question11\cmake-build-debug\question11.exe
43
49
1
进程已结束,退出代码0
```

## 9.心得体会

对这种求最大回报的题目有了更深入的认识与理解。