比较priority\_queue的自定义排序与sort的自定义排序，可以发现，对于实现的都是降序的做法是不同的，sort是使用>，二priority\_queue是使用<

//理解记忆：> 说明大的排在前面，即降序

bool cmp(node x, node y) {

    return x.a > y.a;

}

sort(v.begin(),v.end(),cmp);

priority\_queue的自定义排序:

方法1：运算符重载

struct tmp1{

    int x;

    tmp1(int a) {x = a;}

    bool operator<(const tmp1& a) const {

        return x < a.x; //大顶堆

    }

};

方法2：重写仿函数

struct cmp{

    bool operator() (tmp1 a, tmp1 b){

        return a.x < b.x; //大顶堆

    }

};

priority\_queue<int,vector<int>,cmp> q;

sort(tmp.begin(), tmp.end(), greater<int>()) ;从大到小排列

sort(tmp.begin(), tmp.end(), less<int>());从小到大排列

//降序队列，大顶堆，默认方式

//理解记忆：less是越来越小的意思，即降序，队头的数是最大的数，即大顶堆

priority\_queue <int,vector<int>,less<int> >q;

//升序队列，小顶堆，非默认方式

//理解记忆：greater是越来越大的意思，即升序，队头的数是最小的数，即小顶堆

priority\_queue <int,vector<int>,greater<int> > q;