

E题题解：cdq分治+sort排序+树状数组

题意：求动态逆序对

可以把每个卡片卡成一个三元有序对(t, pos, val)

其中 t 是每个卡片抽到的时间,

pos 是每个卡片在仓库中的位置,

val 是每个卡片的价值。

那么我们要求的就是，在 $1 \sim n$ 时间内，找到这样两个卡片的pairs,

满足 $t_1 < t_2 \&\& pos_1 < pos_2 \&\& val_1 > val_2$ 或者 $t_1 < t_2 \&\& pos_1 > pos_2 \&\& val_1 < val_2$

算法步骤：

- 按照时间顺序，把 pos 小于中间值的放在左边，把 pos 大于中间值的放在右边。这样右边的卡片无论怎样都不会有左边卡片有什么影响。

```
int mid=(l+r)/2;
for(int i=l;i<=r;++i){
    temp[i]=card[i];
}
int ll=l,rr=mid+1,num=0;
for(int i=l;i<=r;++i){
    if(temp[i].Time<=mid) card[ll++]=temp[i];
    else card[rr++]=temp[i];
}
```

- 先正向求左边区间对逆序对的贡献，满足逆序对条件就存入树状数组中；

```
for(ll=l,rr=mid+1;ll<=mid||rr<=r;){
    if(rr>r||ll<=mid&card[ll].pos<card[rr].pos){
        update(card[ll].val,1);//+1代表这个数存在
        ++ll;
        ++num;
    }
    else{
        ans[card[rr].Time]+=num-get_sum(0,card[rr].val);
        ++rr;
    }
}
```

正向贡献求完后清空树状数组，再逆向求左区间对逆序对的贡献；

然后再清空数组；

```
for(int i=l;i<=mid;++i){
    update(card[i].val,-1);
}
```

分别进入左右两个子区间，重复上述操作，直到区间缩为一个点时。

```
//cdq分治大致结构
void cdq(int l,int r){
    if(l>=r) return;
    .....
    cdq(l,mid);
    cdq(mid+1,r);
}
```