

交换排序 NKOJ2180

给出一个长度为 n ($n < 500,000$) 的整数数列，你每一次可以随意交换其中任意**相邻**两个数字的位置。问你至少交换几次，才能使得这个数列是个单调递增数列。

例如有数列：9 1 0 5 4

交换排序后：0 1 4 5 9

只能交换相邻两个数： 交换一次记录一次就可以了

n 的范围是50万，直接暴力枚举， $O(n^2)$ 的复杂度肯定超时

来分析样例数据：9 1 0 5 4

9后面却有4个比9小的元素， 因此9要交换的次数为4

1后面只有1个比1小的元素0， 因此1要交换的次数为1

0后面不存在比他小的元素， 因此0要交换的次数为0

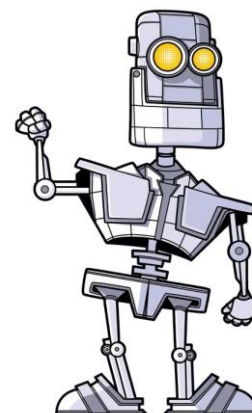
5后面存在1个比他小的元素4， 因此5要交换的次数为1

4是序列的最后元素， 因此4要交换的次数为0

$4 + 1 + 0 + 1 + 0 = 6$ ， 恰恰就是需要的交换次数

求每个数右边有多少个比它小的数字！

问题变成了求数列中总共有多少对**逆序数**！



```
void merge(int L,int Mid,int R)
```

```
{
```

```
    int i=L, j=Mid+1, k=L;
```

```
    while(i<=Mid&& j<=R)
```

```
    {
```

```
        if(s1[i]<=s1[j])s2[k++]=s1[i++];
```

```
        else
```

```
        {
```

```
            cnt+=Mid-i+1
```

```
            s2[k++]=s1[j++];
```

```
        }
```

```
    }
```

```
    while(i<=Mid)s2[k++]=s1[i++];
```

```
    while(j<=R) s2[k++]=s1[j++];
```

```
    for(i=L;i<=R;i++)s1[i]=s2[i];
```

```
}
```

