

### 归并算法案例-逆序对问题

输入一个序列 $\{a_1, a_2, \dots, a_n\}$ ，交换任意两个相邻元素，不超过  $k$  次。交换之后，问最少的逆序对有多少个。

序列中的一个逆序对，是指存在两个数 $a_i$ 和 $a_j$ ，有 $a_i > a_j$ 且 $1 \leq i < j \leq n$ 。也就是说，大的数排在小的数前面。

输入：输入包含多个测试。对于每个测试：

第一行是  $n$  和  $k$ ,  $1 \leq n \leq 10^5, 0 \leq k \leq 10^9$ ; 第二行包括  $n$  个整数 $\{a_1, a_2, \dots, a_n\}$ ,  $0 \leq a_i \leq 10^9$ 。

输出：最少的逆序对数量。

Sample Input:

```
3 1
2 2 1
3 0
2 2 1
```

Sample Output:

```
1
2
```

参考代码:

```
#include <bits/stdc++.h>
const int MAXN = 100005;
typedef long long ll;
ll a[MAXN], cnt;

void Merge(ll a[], ll left, ll mid, ll right){
    ll b[MAXN], i=left, j = mid+ 1, t=left;
    while(i <= mid && j<=right){
        if(a[i]> a[j]){
            b[t++] = a[j++];
            cnt+=mid-i+1;//记录逆序对数量
        }
        else b[t++]=a[i++];
    }
    //一个子序列中的数都处理完了， 另一个还没有， 把剩下的直接复制过来
    while(i <= mid) b[t++]=a[i++];
    while(j <= right) b[t++]=a[j++];
    for(i=left; i<=right; i++) a[i] = b[i];//把排好序的 b[]复制回 a[]
}

void Mergesort(ll a[], ll left, ll right){
    if(left<right){
        ll mid = left+ (right-left)/2;//即 mid =(left+ right)/2 平分成两个子序列,避免 left+
        right 和溢出
        Mergesort(a, left, mid);
```

```

        Mergesort(a, mid+ 1, right);
        Merge(a, left, mid, right);//合并
    }
}

int main(){
    ll n,k;
    while(scanf("%lld%lld", &n, &k)==2){
        cnt = 0;
        for(ll i=0;i<n;i++) scanf("%lld", &a[i]);
        Mergesort(a,0,n- 1);//归并排序
        if(cnt<=k) printf("0\n");
        else      printf("%lld\n", cnt - k);
    }
    return 0;
}

```