NlogN求逆序数对

|  |
| --- |
| #include<iostream>  #include<stdio.h>  #include<string.h>  #include<algorithm>  #include<cmath>  using namespace std;  const int maxn=1e5+5;  int is1[maxn],is2[maxn];// is1为原数组，is2为临时数组，n为个人定义的长度  long long merge(int low,int mid,int high)  {  int i=low,j=mid+1,k=low;  long long count=0;  while(i<=mid&&j<=high)  if(is1[i]<=is1[j])// 此处为稳定排序的关键，不能用小于  is2[k++]=is1[i++];  else  {  is2[k++]=is1[j++];  count+=j-k;// 每当后段的数组元素提前时，记录提前的距离  }  while(i<=mid)  is2[k++]=is1[i++];  while(j<=high)  is2[k++]=is1[j++];  for(i=low;i<=high;i++)// 写回原数组  is1[i]=is2[i];  return count;  }  long long mergeSort(int a,int b)// 下标，例如数组int is[5]，全部排序的调用为mergeSort(0,4)  {  if(a<b)  {  int mid=(a+b)/2;  long long count=0;  count+=mergeSort(a,mid);  count+=mergeSort(mid+1,b);  count+=merge(a,mid,b);  return count;  }  return 0;  }  int main()  {  int n,x,y;  while(~scanf("%d%d%d",&n,&x,&y))  {  int tmp=min(x,y);  for(int i=1;i<=n;i++)  scanf("%d",&is1[i]);  printf("%lld\n",tmp\*mergeSort(1,n));  }  return 0;  } |