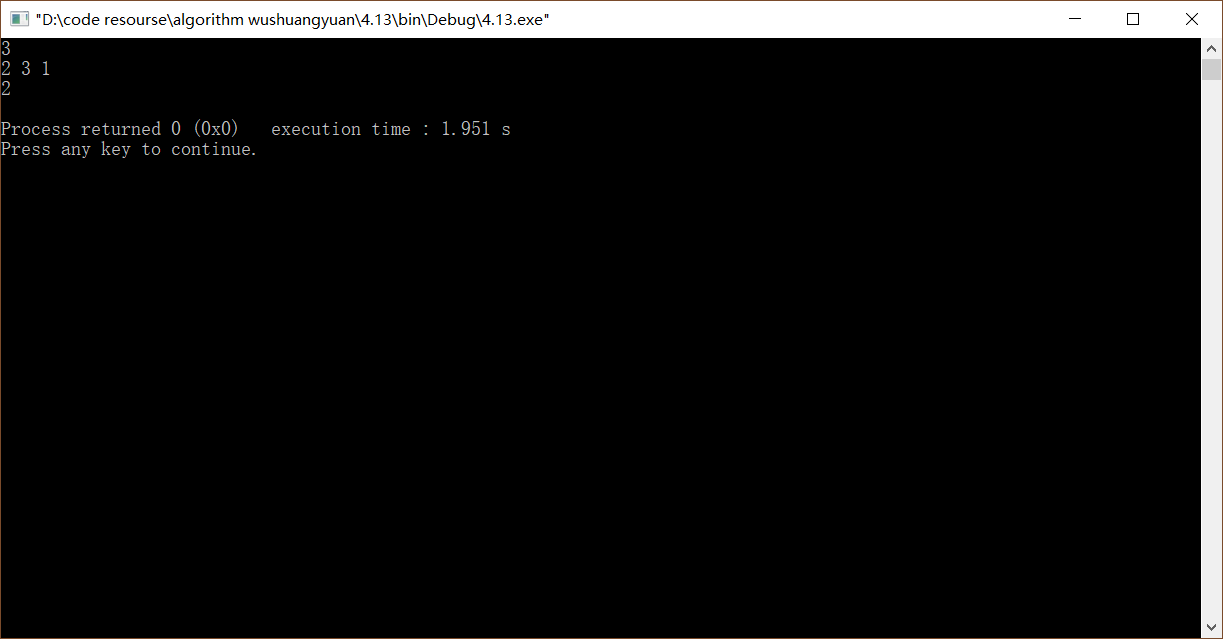
习题四 13.

算法思路：

对整个数列递归地划分子序列，直到序列只含一个元素，再分别递归地合并，这会使每个元素都遇到这个序列中的其他每一个元素，所以不会产生遗漏。

运行截图：



由归并排序的复杂度可知，该算法复杂度为o(nlogn).

代码：

#include <iostream>

#include <vector>

using namespace std;

int merge(vector<int>&a, int l, int mid, int r)

{

int count = 0;

int p1 = l;

int p2 = mid + 1;

int i = 0;

vector<int>help(r - l + 1);

while (p1 <= mid && p2 <= r)

{

count += a[p1] > a[p2] ? (mid - p1 + 1) : 0;

help[i++] = a[p1] <= a[p2] ? a[p1++] : a[p2++];

}

while (p1 <= mid)

help[i++] = a[p1++];

while (p2 <= r)

help[i++] = a[p2++];

for (i = 0; i < (int)help.size(); i++)

a[l + i] = help[i];

return count;

}

int mergeSort(vector<int>&a, int l, int r)

{

if (l == r)

return 0;

int mid = l + (r - l) / 2;

return mergeSort(a, l, mid) + mergeSort(a, mid + 1, r) + merge(a, l, mid, r);

}

int SmallSum(vector<int>a)

{

if (a.size() < 2)

return 0;

return mergeSort(a, 0, a.size() - 1);

}

int main()

{

vector<int> a;

int n,tc;

cin>>n;

for (int i=0;i<n;i++) {

cin>>tc;

a.push\_back(tc);

}

cout << SmallSum(a) << endl;

return 0;

}