1089. Insert or Merge (25)

时间限制

200 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

According to Wikipedia:

**Insertion sort** iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.

**Merge sort** works as follows: Divide the unsorted list into N sublists, each containing 1 element (a list of 1 element is considered sorted). Then repeatedly merge two adjacent sublists to produce new sorted sublists until there is only 1 sublist remaining.

Now given the initial sequence of integers, together with a sequence which is a result of several iterations of some sorting method, can you tell which sorting method we are using?

**Input Specification:**

Each input file contains one test case. For each case, the first line gives a positive integer N (<=100). Then in the next line, N integers are given as the initial sequence. The last line contains the partially sorted sequence of the N numbers. It is assumed that the target sequence is always ascending. All the numbers in a line are separated by a space.

**Output Specification:**

For each test case, print in the first line either "Insertion Sort" or "Merge Sort" to indicate the method used to obtain the partial result. Then run this method for one more iteration and output in the second line the resulting sequence. It is guaranteed that the answer is unique for each test case. All the numbers in a line must be separated by a space, and there must be no extra space at the end of the line.

**Sample Input 1:**

10

3 1 2 8 7 5 9 4 6 0

1 2 3 7 8 5 9 4 6 0

**Sample Output 1:**

Insertion Sort

1 2 3 5 7 8 9 4 6 0

**Sample Input 2:**

10

3 1 2 8 7 5 9 4 0 6

1 3 2 8 5 7 4 9 0 6

**Sample Output 2:**

Merge Sort

1 2 3 8 4 5 7 9 0 6

[提交代码](https://www.patest.cn/contests/pat-a-practise/1089)

当时直接改的话估计还不一定能改过来。。。

第一个是insertion sort的坑……

第二是merge sort要对末尾不在base覆盖范围内的函数进行排序……

我记得当初应该看的时候没看到这个要求……

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#define MAX 505

using namespace std;

vector<int>Insert\_sequence;

vector<int>processing;

vector<int>Merge\_sequence;

vector<int>origin\_sequence;

vector<int>target\_sequence;

int base = 2;

int pos = 2;

void Insert()

{

if (pos <= Insert\_sequence.size())

{

sort(Insert\_sequence.begin(), Insert\_sequence.begin() + pos);

pos++;

}

return;

}

void Merge()

{

int now\_pos = 0;

if (base >= Merge\_sequence.size())

{

sort(Merge\_sequence.begin(), Merge\_sequence.end());

return;

}

else

{

while (now\_pos + base <= Merge\_sequence.size())

{

sort(Merge\_sequence.begin()+now\_pos, Merge\_sequence.begin() + now\_pos + base);

now\_pos += base;

}

sort(Merge\_sequence.begin() + now\_pos, Merge\_sequence.end());

base \*= 2;

return;

}

}

void output(vector<int>res)

{

printf("%d", res[0]);

for (int i = 1; i < res.size(); i++)

{

printf(" %d", res[i]);

}

}

int main()

{

int num;

cin >> num;

for (int i = 0; i < num; i++)

{

int temp;

scanf("%d", &temp);

origin\_sequence.push\_back(temp);

}

for (int i = 0; i < num; i++)

{

int temp;

scanf("%d", &temp);

target\_sequence.push\_back(temp);

}

Insert\_sequence = origin\_sequence;

Merge\_sequence = origin\_sequence;

vector<int>result = origin\_sequence;

sort(result.begin(), result.end());

while (1)

{

Insert();

if (target\_sequence==Insert\_sequence)

{

cout << "Insertion Sort" << endl;

Insert();

output(Insert\_sequence);

return 0;

}

if (result == Insert\_sequence)

break;

}

while (1)

{

Merge();

if (target\_sequence == Merge\_sequence)

{

cout << "Merge Sort" << endl;

Merge();

output(Merge\_sequence);

return 0;

}

}

}