LCS转LIS的做法

题目要求要nlogn求LCS，显然通常的LCS是n^2的，所以需要转换到LIS，用LIS的nlogn方法求

怎么转换呢，首先将第一个数组进行离散化，将第一组数据与它的位置对应上

比如 3-1， 2-2， 1-3，4-4，5-5

这样每个数字的位置都是上升的，然后我们根据第2个数列

第2个数列中排在前面的对应到第一个数组的下标肯定更小，这样就转化成了求最长上升子序列了（当然如果对应不到第一个数组就说明不是公共元素了）

每个数字的位置都是上升的， 然后我们根据第二个序列， 从头开始判断 第二个序列的每一个数字在离散化处理好的数组中

的位置，强制使它有序，就转化成了 nlongn复杂度求LIS的问题

#include <cstdio>

#include<iostream>

#include<vector>

#include<algorithm>

using namespace std;

#define ll long long

#define INF 1000000007

int n;

int a[100010],b[100010];

int f[100010];

int main()

{

//freopen("input.txt","r",stdin);

scanf("%d",&n);

int x;

for(int i=1;i<=n;++i)

scanf("%d",&x),a[x]=i;

for(int i=1;i<=n;++i)

scanf("%d",&b[i]);

for(int i=0;i<=100010;++i)

f[i]=INF;

for(int i=1;i<=n;i++)

{

int pos=lower\_bound(f,f+1+n,a[b[i]])-f;

f[pos]=a[b[i]];

}

int ans=0;

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

if(f[i]!=INF)

++ans;

printf("%d\n",ans);

return 0;

}