**Cdq fenzhi**

**对于序列A，它的逆序对数定义为满足*i*<*j*，且A*i*>A*j*的数对(*i*,*j*)的个数。给1到*n*的一个排列，按照某种顺序依次删除*m*个元素，你的任务是在每次删除一个元素之前统计整个序列的逆序对数。**

**Input**

**输入第一行包含两个整数*n*和*m*，即初始元素的个数和删除的元素个数。以下*n*行每行包含一个1到*n*之间的正整数，即初始排列。以下*m*行每行一个正整数，依次为每次删除的元素。**

**Output**

**输出包含*m*行，依次为删除每个元素之前，逆序对的个数。**

**Sample Input**

**5 4**

**1**

**5**

**3**

**4**

**2**

**5**

**1**

**4**

**2**

**Sample Output**

**5**

**2**

**2**

**1**

**样例解释**

**(1,5,3,4,2)◊(1,3,4,2)◊(3,4,2)◊(3,2)◊(3)。**

**HINT**

**N<=100000 M<=50000**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem: 3295**

**User: ccs1997**

**Language: C++**

**Result: Accepted**

**Time:1592 ms**

**Memory:4788 kb**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//**[**http://spade47.cr0.cn/blog/archives/programming/cqoi2011-inverse**](http://spade47.cr0.cn/blog/archives/programming/cqoi2011-inverse)

**#include <cstdio>**

**#include <iostream>**

**#define MAXN 100009**

**using namespace std;**

**int a[ MAXN ], b[ MAXN ], c[ MAXN ], x[ MAXN ], y[ MAXN ], o[ MAXN ], ans[ MAXN ];**

**int ll, rr, N, M, t;**

**long long ans2[ MAXN ];**

**void Work(int l, int r)//已经保证添加顺序从大到小**

**{**

**if (l==r) return ;**

**int m = (l+r)>>1;**

**Work(l, m);**

**Work(m+1, r);**

**t++;**

**ll = l, rr = m+1;//指针**

**for (int i=l; i<=r; i++)**

**if ((ll<=m && a[x[ll]]<a[x[rr]]) || rr>r)**

**//以x[]映射到a[]的大小排序 保证数值从小到大**

**{**

**y[i] = x[ll];//暂存**

**int j = b[y[i]];//位置**

**while (j>0)**

**{**

**if (o[j]!=t) o[j] = t, c[j] = 0;//不在本层则清空**

**c[j]++;**

**j -= j&(-j);//pos向前修改**

**}**

**ll++;**

**}**

**else**

**{**

**y[i] = x[rr];**

**int j = b[y[i]];**

**while (j<=N)**

**{**

**if (o[j]==t) ans[y[i]] += c[j];**

**j += j&(-j);//pos向后统计 保证所统计的pos比当前pos大**

**}**

**rr++;**

**}**

**for (int i=l; i<=r; i++) x[i] = y[i];**

**return ;**

**}**

**int main()**

**{**

**scanf("%d %d", &N, &M);**

**for (int i=1; i<=N; i++) scanf("%d", &x[i]);**

**for (int i=1; i<=N; i++) y[x[i]] = i;**

**int i;**

**for (i=1; i<=M; i++)**

**{**

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

**o[a[i]] = -1;//flag**

**}**

**for (int j=1; j<=N; j++) if (!o[j]) a[i++] = j;//加上未被删除的元素**

**for (int i=1; i<=N; i++) b[i] = y[a[i]];//第i个要删除的元素的pos**

**for (int i=1; i<=N; i++) x[i] = N-i+1;//第i个元素的添加顺序大小**

**Work(1, N);**

**for (int i=1; i<=N; i++)**

**a[i] = N-a[i]+1, b[i] = N-b[i]+1, x[i] = N-i+1;**

**Work(1, N);**

**for (int i=N; i>0; i--) ans2[i] = ans2[i+1]+ans[i];**

**for (int i=1; i<=M; i++) printf("%lld\n", ans2[i]);**

**return 0;**

**}**