C1-B

题目内容

题目描述

Zhoues 很喜欢研读《算法导论》，有一天在暗中观察该书第 24 页的时候，想到了一个问题并需要你来帮他解决，问题如下：

- 给定一个序列 $a_1, a_2, ..., a_n$ 和一个正整数 k，如果 $1 \le i < j \le n$ 且 $a\_i > k \cdot a\_j$ 我们就将 (i, j) 称作一个**逆序 k 倍对**。请你计算序列中逆序 k 倍对的个数。

输入格式

第一行两个正整数 n, k （$1 \le n \le 10^5$，$1 \le k \le 10$），含义同题目描述。

第二个 n 个正整数 $a_1, a_2, ..., a_n$ （$1 \le a_i < 2^{31}$），含义同题目描述。

对于得分占比 10% 的测试点，保证 $1 \le n \le 100$。

输出格式

一行一个非负整数，表示序列中逆序 k 倍对的个数。

输入样例

```
5 2
5 4 3 2 1
```

输出样例

```
4
```

题解思路

该题目的灵感来源于《算法导论》第二章课后习题 2-4——逆序对，即使用归并排序（分治）对该序列进行排序的时候，对满足逆序对的条件的两个数进行计数。

这里不细讲如何求逆序对，具体求法（归并排序，离散化后树状数组等等）可以自行查阅资料，这里以归并排序为例。

在归并排序的过程中，假设对于数组 $a[l..r]$ 而言，我们已经分别求出了子数组 $a[l..m]$与 $a[m+1..r]$ 的逆序 k 倍对数目，并已将两个子数组分别排好序，则 $a[l..r]$中的逆序 k 倍对数目，就等于两个子数组的逆序 k 倍对数目之和，加上左右端点分别位于两个子数组的逆序 k 倍对数目。

我们可以为两个数组分别维护指针i, j。对于任意给定的 i 而言，我们不断地向右移动j，直到 $a[i] \leq k \cdot a[j]$。此时，意味着以i 为左端点的逆序 k 倍对数量为$j-m-1$。随后，我们再将 i 向右移动一个单位，并用相同的方式计算以 i 为左端点的逆序 k 倍对数量。不断重复这样的过程，就能够求出所有左右端点分别位于两个子数组的逆序 k 倍对数目。

参考代码

```
#include<stdio.h>
int a[1000002], n, k, num;

long long reversePairsRecursive(int* nums, int left, int right) {
    if (left == right) {
        return 0;
    } else {
        int mid = (left + right) / 2;
        int n1 = reversePairsRecursive(nums, left, mid);
        int n2 = reversePairsRecursive(nums, mid + 1, right);
        long long ret = n1 + n2;

        int i = left;
        int j = mid + 1;
        while (i <= mid) {
            while (j <= right && (long long)nums[i] > k * (long long)nums[j])
j++;
            ret += (j - mid - 1) ;
            i++;
        }

        int sorted[right - left + 1];
        int p1 = left, p2 = mid + 1;
        int p = 0;
        while (p1 <= mid || p2 <= right) {
            if (p1 > mid) {
                sorted[p++] = nums[p2++];
            } else if (p2 > right) {
                sorted[p++] = nums[p1++];
            } else {
                if (nums[p1] < nums[p2]) {
                    sorted[p++] = nums[p1++];
                } else {
                    sorted[p++] = nums[p2++];
                }
            }
        }
        for (int i = 0; i < right - left + 1; i++) {
            nums[left + i] = sorted[i];
        }
        return ret;
```

```
    }
}

int reversePairs(int* nums, int numsSize) {
    if (numsSize == 0) {
        return 0;
    }
    return reversePairsRecursive(nums, 0, numsSize - 1);
}

int main() {
    scanf("%d %d",&n,&k);
    for(int i = 0; i < n ; i++) {
        scanf("%d",&a[i]);
    }
    printf("%lld\n",reversePairs(a,n));
    return 0;
}
```

Author: 周恩申