**A1206. 小Z的袜子**

**时间限制：1.0s   内存限制：512.0MB**

总提交次数：[744](http://www.tsinsen.com/AllSubmits.page?type=a&gpid=A1206,D863,D1309,D1485,D2053,P357)   AC次数：210   平均分：44.44

将本题分享到：

[查看未格式化的试题](http://www.tsinsen.com/A1206###)   [提交](http://www.tsinsen.com/A1206###)   [试题讨论](http://www.tsinsen.com/A1206###)

**试题来源**

　　2010中国国家集训队命题答辩

**问题描述**

　　作为一个生活散漫的人，小Z每天早上都要耗费很久从一堆五颜六色的袜子中找出一双来穿。终于有一天，小Z再也无法忍受这恼人的找袜子过程，于是他决定听天由命……  
　　具体来说，小Z把这N只袜子从1到N编号，然后从编号L到R(L 尽管小Z并不在意两只袜子是不是完整的一双，甚至不在意两只袜子是否一左一右，他却很在意袜子的颜色，毕竟穿两只不同色的袜子会很尴尬。  
　　你的任务便是告诉小Z，他有多大的概率抽到两只颜色相同的袜子。当然，小Z希望这个概率尽量高，所以他可能会询问多个(L,R)以方便自己选择。

**输入格式**

　　输入文件第一行包含两个正整数N和M。N为袜子的数量，M为小Z所提的询问的数量。  
　　接下来一行包含N个正整数Ci，其中Ci表示第i只袜子的颜色，相同的颜色用相同的数字表示。  
　　再接下来M行，每行两个正整数L，R表示一个询问。

**输出格式**

　　输出文件包含M行，对于每个询问在一行中输出分数A/B表示从该询问的区间[L,R]中随机抽出两只袜子颜色相同的概率。若该概率为0则输出0/1，否则输出的A/B必须为最简分数。（详见样例）

**样例输入**

6 4  
1 2 3 3 3 2  
2 6  
1 3  
3 5  
1 6

**样例输出**

2/5  
0/1  
1/1  
4/15

**样例说明**

　　询问1：共C(5,2)=10种可能，其中抽出两个2有1种可能，抽出两个3有3种可能，概率为(1+3)/10=4/10=2/5。  
　　询问2：共C(3,2)=3种可能，无法抽到颜色相同的袜子，概率为0/3=0/1。  
　　询问3：共C(3,2)=3种可能，均为抽出两个3，概率为3/3=1/1。  
　　注：上述C(a, b)表示组合数，组合数C(a, b)等价于在a个不同的物品中选取b个的选取方案数。

**数据规模和约定**

　　30%的数据中 N,M ≤ 5000；  
　　60%的数据中 N,M ≤ 25000；  
　　100%的数据中 N,M ≤ 50000，1 ≤ L < R ≤ N，Ci ≤ N。

题目链接[点击打开链接](http://www.tsinsen.com/A1206)

题意：

中文题目就不解释了。

思路：

对于L,R的询问。设其中颜色为x,y,z....的袜子的个数为a,b,c。。。

那么答案即为(a\*(a-1)/2+b\*(b-1)/2+c\*(c-1)/2....)/((R-L+1)\*(R-L)/2)

化简得:(a^2+b^2+c^2+...x^2-(a+b+c+d+.....))/((R-L+1)\*(R-L))

即：(a^2+b^2+c^2+...x^2-(R-L+1))/((R-L+1)\*(R-L))

所以这道题目的关键是求一个区间内每种颜色数目的平方和。

但问题时怎么快速求解呢？

对于一般区间维护类问题一般想到用线段树。但是这题完全不知道线段树怎么做，百度了下。知道是莫队算法。

于是乎学习了下。写写学习的心得吧。

莫队算法是莫涛发明了。感觉这人蛮牛逼的。但是网上各种百度他的论文却找不到了。只好到别人的博客里学习学习。莫队算法是离线处理一类区间不修改查询类问题的算法。就是如果你知道了[L,R]的答案。你可以在O(1)的时间下得到[L,R-1]和[L,R+1]和[L-1,R]和[L+1,R]的答案的话。就可以使用莫队算法。

对于莫队算法我感觉就是暴力。只是预先知道了所有的询问。可以合理的组织计算每个询问的顺序以此来降低复杂度。要知道我们算完[L,R]的答案后现在要算[L',R']的答案。由于可以在O(1)的时间下得到[L,R-1]和[L,R+1]和[L-1,R]和[L+1,R]的答案.所以计算[L',R']的答案花的时间为|L-L'|+|R-R'|。如果把询问[L,R]看做平面上的点a(L,R).询问[L',R']看做点b(L',R')的话。那么时间开销就为两点的曼哈顿距离。所以对于每个询问看做一个点。我们要按一定顺序计算每个值。那开销就为曼哈顿距离的和。要计算到每个点。那么路径至少是一棵树。所以问题就变成了求二维平面的最小曼哈顿距离生成树。

关于二维平面最小曼哈顿距离生成树。感兴趣的可以参考[点击打开链接](http://blog.csdn.net/huzecong/article/details/8576908)

这样只要顺着树边计算一次就ok了。可以证明时间复杂度为n\*sqrt(n)这个我不会证明。

但是这种方法编程复杂度稍微高了一点。所以有一个比较优雅的替代品。那就是先对序列分块。然后对于所有询问按照L所在块的大小排序。如果一样再按照R排序。然后按照排序后的顺序计算。为什么这样计算就可以降低复杂度呢。

一、i与i+1在同一块内，r单调递增，所以r是O(n)的。由于有n^0.5块,所以这一部分时间复杂度是n^1.5。  
二、i与i+1跨越一块，r最多变化n，由于有n^0.5块，所以这一部分时间复杂度是n^1.5  
三、i与i+1在同一块内时变化不超过n^0.5，跨越一块也不会超过2\*n^0.5，不妨看作是n^0.5。由于有n个数，所以时间复杂度是n^1.5  
于是就变成了O(n^1.5)了。