2016年西安电子科技大学ACM校赛简要解题报告

Written by mathlover

A题-缩水的面包

手速签到题

直接输出$n-10$即可

B题-万神的方程

关键字：分类讨论

注意到题目并没有保证这是个二次函数，因此a是可以为0的，再加上题目提示了会有无限多个根的情况，无限多个根当且仅当a=b=c=0

因此只要分类讨论即可

$a=0\ b=0\ c=0 \Rightarrow inf$

$a=0\ b=0\ c\neq 0 \Rightarrow 0$

$a=0\ b\neq 0 \Rightarrow 1$(一次函数)

$a\neq 0\ \Delta>0 \Rightarrow 2$

$a\neq 0\ \Delta=0 \Rightarrow 1$

$a\neq 0\ \Delta<0 \Rightarrow 0$

C题-数据库查询

关键字：模拟

题目数据范围并不大，因此可以不用任何数据结构进行优化，直接设计一个struct结构体或者class类，存储姓名学号以及课程，注意输入输出的格式。并且输出前需要按学号关键字进行排序。冒泡，快速排序都是允许的。

另外一种方法是，注意到学号不超过10^6，一次直接使用一个10^6大小的静态数组，以学号为下标进行存储。输出时按学号从小到大遍历，把名字为空的跳过，输出就行了。

D题-万神的线段

关键字：计算几何，分类讨论

首先可以特判垂直于x轴(斜率不存在)的线段，两两平行或重合。 然后其余的线段可以按斜率进行排序，然后分别从小到大分别统计每个斜率出现了多少次。

把两类的答案加起来就是最终答案。

计算几何不卡精度就是好。

E题-Nhywieza 的串

关键字：贪心

考虑到每次翻转都是连续的子串，因此只要统计有多少组连续的0即可，每一组连续的0都要消耗一次翻转的次数。

F题-蛇形矩阵

关键字：找规律

直接粗暴地生成整个矩阵是不现实的，因此我们需要观察一下这个矩阵的一些性质。

首先我们注意到蛇形矩阵可以分成$\frac{k+1}{2}$层，其中第p层($p = 1, \ldots, \frac{k+1}{2}$)的第一个数字(左上角)是$4(p-1)(k-p+1)+1$.左上角的数的坐标是$(p,p)$.如果$k$是奇数的话，那么最后一层只有一个数是$k^2$.

因此如果我们给定x,y，我们首先判断这个数是在第几层，判断方法也是很明显的，看这个数距离矩阵的四条边的哪一边最近。那么这个距离就是这个数所在的层数。即$min(x, k+1-x, y, k+1-y)$

然后我们又可以把一层沿着主对角线分成两半，其中第一半是从左往右递增，从上到下递增，另一半是从左往右递减，从上到下递减。我们也可以根据x,y大小关系来判断是在哪一半。

最后我们只要计算一下我们找的数与该层左上角的数(p,p)的曼哈顿距离$(x-p)+(y-p)$，即可知道我们这个数于左上角的数之间相差多少。

G题-等待队列

关键字：单调队列

我们可以假设把所有的人都是从一开始第一个时间点$t=0$进入队列，并且他的不耐烦程度为$x' = x - t$，其中t为他实际加入的时间点。

这样当我们进行操作3查询时，只需要用当前的时间点$t' + x'$即可得到$x' + t' = x + (t' - t)$就是我们要求的答案。

然后操作3需要我们维护队列中的最大值，因此我们可以使用单调队列去维护这个队列。使这个队列中的不耐烦值总是递减的。这样我们每次从这个队列中查找最大值，只需要找到队首元素即可。

当执行操作1时，我们需要把新来的不耐烦值x(已经减去当前的时间点)加入到单调队列中，我们从队尾开始往队头进行比较，直到找到一个比x大的数，把后面的全部移出队列，把新来的x就放在这里。

当执行操作2时，我们不能像普通队列中直接出队，因为实际上队首前面还有一些被我们在入队时已经移除的元素，因此我们要用两个计数器来记录当前入队的编号以及出队的编号，当真正轮到队首元素出队时，才真正执行pop操作。

当执行操作3时，只需要访问队首元素即可。

单调队列的实现可以自己使用一个数组，然后用两个指针或下标记录队首和队尾的位置。

或者使用C++的STL中的双端队列deque，该队列支持`push_back,push_front,pop_back,pop_front`四种操作(从队尾插入，从队首插入，从队尾移出，从队首移出)

实际上在本题不需要用到从队首插入。

由于我们需要记录两个信息：不耐烦值 以及入队时间点，因此队伍里面的元素实际上是一个pair或者自己手写的结构体。

之所以可以这样做，是利用了本题的一个事实：若存在两个人A和B，A比B早入队，在B入队时，A还在队列中且A的不耐烦值比B要少，那么A是永远不会成为不耐烦值最大的人。因此在这种情况下，A是可以忽略的(体现在B入队时就把A从队尾剔除掉了)

H题-Tom的树

关键字：并查集

注意到我们所要求的式子，其实是可以看成是两个子问题：求出C_n^2个路径上的最大值之和，以及求出C_n^2个路径上的最小值之和。这两个值作差就是最终答案。而且这两个问题其实是同一个问题。我们只看第一个问题，求出C_n^2个路径上的最大值之和。

一棵树的生成，可以看成是往一堆离散的点中添加连接其中两个点的边。如果需要维护这些点的连接，我们可以使用一个很常用的数据结构——并查集来维护。 在《算法竞赛入门经典》以及《算法竞赛训练手册》《算法导论》中都有介绍。

生成边的顺序是可以任意改变的，因此我们要求最大值，就让边从小到大生成。保证每次生成的边都是图中最大的边。

每次往图中添加边，我们可以用并查集来连接该边的两个端点A,B，因为我们保证了这个边是当前图中最大的，因此点A代表的子树中的所有点到点B代表的子树的所有点的路径，必然会经过这条边，且这条边是这些路径的最大值。因此这条边的贡献为$size[A] * size[B]$。其中$size[A]$表示以点A为根的子树的体积(即点的数量)。

利用并查集合并两棵子树时，记得要更新子树的$size$大小。

求最小值只需要把边从大到小排序生成即可。

I题-Orz Fatality

关键字：母函数 组合数 找规律 YY

本题需要求的就是 $\sum_{i=0}^{x} C_x^i C_y^{i+k}$

最暴力的方式就是用$O(min(x, y-k))$的时间复杂度枚举i

但是在10^6组极限数据面前，这样会跑上好几天。 因此我们需要化简这个式子。

方法1：打表 写出一个小程序 把$x, y, k \leq 10$的所有答案打到一个文件中一个k对应一个$n \times m$的矩阵 观察可以发现都是组合

数....

$k=0 \Rightarrow C_{x+y}^{x}$

$k=1 \Rightarrow C_{x+y}^{x+1}$

.... 总结规律:

对于任意x, y, k 答案是C_{x+y}^{x+k}

对k使用数学归纳法可以证明

方法2：母函数推导

利用二项式定理

$\sum_{i=0}^{n} C_x^i C_y^{i+k}$ 其实就是$(1+a)^y(1+\frac{1}{a})^x$中的a^k项系数(第一个二项式的a^j系数是C_y^j 第二个二项式的a^{-i}系数是C_x^i，令$j-i=k$,即得$j=i+k$，即$\sum_{i=0}^{n} C_x^i C_y^{i+k}$)

另一方面$(1+a)^y(1+\frac{1}{a})^x = (1+a)^y(1+a)^x a^{-x} = (1+a)^{x+y}a^{-x}$

这里的a^k项系数为C_{x+y}^{x+k}

如果有更好的思路或者有什么差错，请加入ACM_XDU-ICPC群 116225686 联系我