人人公共主页: **http://page.renren.com/601172904?id=601172904**
ACM_DIY QQ 群:
151934295

A. Abor's Problem
计算每个人的成为 "abor" 的期望
假设一个人 x 认识 C 个人,那么首先如果他是"abor", 必须满足
(1) 性别为男
(2) 认识的人中，至少有 m 个 MM
于是这个期望就是
E = (0.5) * pow(0.5, C) * {C(C,m) + C(C,m+1)+...+C(C,C)}
累加即可
复杂度 O(n^2)

B. Closest Number
题意：找距离最近的小于 a[i]的数字，距离相同，选左边的。

用一个类似栈的方法维护就行了。
比如现在从左到右扫，现在栈里是，1，3，4，5， a[i]是 2.那么栈里 3，4，5 弹出，2 进去。
因为右边其他数字，都是 2 离得更近。3，4，5 已经没意义了。
这个方法左边右边各扫描一下就好了。

C.Construct The Graph
利用 2 进制构造
构造形如一条链的图
例如
A=B-C 其方案数为 2 种 (等号代表 2 个边~）
如果要将方案数 乘 2
那么只需要添加一个节点 D
D=A=B-C
如果是 乘 2 以后加 1，那么是
D=A=B-C
|_____|

其中 D->C 的距离等于最短路(D->A->B->C)
于是需要用 logn 个节点和 logn 条边~

D.Crop Circle
本题利用了圆和圆之间的关系:
如果一个圆套住了另外一个圆，那么可以等价为一个树上的一条边！
于是本题的 m 条信息是给定了度数的信息，
(0) 如果给出的 y 值不合理，则不存在;
(1) 当给出的信息矛盾时，则不存在;

(2) 由于最终的结果可能是多棵(子)树,因此将外面的无穷大的平面看成了一个新的“大圆”。于是新的树拥有 n + 1 个节点;
(3) 我们知道 n+1 个节点的树，其每个节点的度数和为 2(n+1)-2=2n, 而最终 “大圆” 的度数必须严格大于 0，同时不能太小！
我们知道“大圆” 的度数就是无穷大平面数下去能数到几个圆，对于每一个合法的 y,由于这 y 个圆都在某个圆里面，所以这 y 个圆肯定不会在“大圆” 内。
于是就可以得到最终“大圆” 的度数的一个区间。判断即可

E.How Many Nodes
假设每个线段树的节点被覆盖作为事件 I_i, 其中 i 为节点的编号，那么
E(节点被覆盖的数目) = sigma{ E(I_i) }
下面考虑 E(I_i)的计算方法:
(1) 对于一个节点[l,r]，如果它是根，则只可能有一种，那就是查询全区间才能覆盖它；
(2) 对于一个节点[l,r],如果它是某个节点的左孩子,那么覆盖这个节点的区间:
(a) 左端点随便选, 0~l 都可以 (b) 右端点可以选择 r~Fr-1, 其中 Fr 是它的父亲的右端点(如果选了父亲右端点以及其右边的，那么就覆盖了它的父亲，就不是覆盖这个节点了~)
(3) 对于一个节点[l,r],如果它是某个节点的右孩子，那么覆盖这个节点的区间:
(a) 左端点选, Fl+1~l 都可以,Fl 是其父亲的左端点(如果选了父亲左端点以及其左边的，那么就覆盖了它的父亲，就不是覆盖这个节点了~) (b) 右端点可以选择 r~n

F. Lhxsb
首先采用倒序思想，则删除点就变为添加点.

方法 1：
注意到每个点往右的视角其实被它右边的点形成的上凸包所限制（左边同理），那么我们用线段树维护凸包。其中线段树代表[l,r]的节点就用平衡树维护[l,r]的凸包。
对于每次添加操作，在 O(logN)个线段树节点中维护新的凸包，每次维护还需要 O(logN)的二分，复杂度 O(log^2N)。
对于每次询问操作，在点的右边（左边同理）的 O(logN)凸包中对于每个凸包都二分答案，最后取最优值，复杂度 O(log^2N)。
总复杂度 O(NlogN+Qlog^2N)。（标程）

方法 2：
对于时间分治，把问题转化为静态版。此时从左向右扫描，用栈维护即可。
最终复杂度 O(N+Qlog^2N)。

G.MultiplicationTable
显然要求的是一个 Abel 群。根据 Abel 群分解定理，每个有限 Abel 群均能分解成一些循环子群的直和。
由于本题的附加条件非常松，我们可以认为对于所有 n，均存在与 n 阶循环群同构的答案。
然后就是要求一个映射。这个就可以搜了（因为我不知道我搜的方法对不对，所以这里就不说了 ==）。

n 为一些特殊值时候的构造（如果不知道，不影响做出本题）:

1. n+1 为素数 这个时候直接令 i*j=ij%(n+1)即可。
2. 2n+1 为素数 这个时候直接令 i*j=min(ij%(2n+1),2n+1-ij%(2n+1))即可。
证明均是简单的。
可能 n 为其他值的也能构造，不过我不会。

搜索的话对于 34，37，38，跑得比较慢。这个打表就可以了。当然也可以手造。如果知道只要构造出一个与循环群同构的群，手造速度还是非常快的。可以使用程序配合手造。
我的非打表程序对于 40 个点总共跑了不到 1 分钟，代码不到 2KB。

H.Number Theory
n^(2n)+n^m=a(mod p)

若 m=0，则 n^m=1, n^(2n)=a-1 (mod p)
若 a-1 不是模 p 的二次剩余，则显然无解；
否则我们只要解 n^n=sqrt(a-1) (mod p)
若 a-1=0，那么 n=0 (mod p)；
否则令 n0=p 的某个原根，解出 n0^x=sqrt(a-1) (mod p)
这样我们有方程组 n=n0 (mod p)和 n=x(mod (p-1))，CRT 合并即可

若 m>=1，我们随机原根 n0，直到 a-n0^m 是模 p 的二次剩余，然后可以用之前的方法解；
我们来考虑有解的概率 (这段纯粹的 YY，完全不严谨)
是否二次剩余我们大致可以看成概率 1/2。
对于每个 n0，x^m=n0^m 的解在模 p 的剩余系内不超过 m 个。
而 p 的原根有恰好 phi(p-1)个
因此我们可以得到 phi(p-1)/m 个不同的数，
它们都不是模 p 的二次剩余的概率是(1/2)^(phi(p-1)/m)
phi(p-1)>p/4
因此在 p 很大的时候，几乎可以保证总是有解
我找到的最大的 m>=1 的无解情况，p 是 241

I.Sequence Transformation
定义 S_i = sigma{ A[i]}
那么每次操作，就是对于 S[] 这个数组的 相邻 2 个元素(这 2 个元素表示的是和)做交换操作
于是判定性问题很容易的可以通过对 A,B 的和 S 数组判定是否相同(前提是 2 个数组的和相同)

J.Shortest Path
题意：给定 n 个点，m 条边，求最短的经过所有边至少一次的路径。

说白了，该题的意思就是一笔划通过所有边至少一次。先统计所有的点的度数并累加所有边，所有的偶数度节点不必管，奇数度节点额外考虑。因为一笔划的情况奇数度节点只会有 0 个或者 2 个。所以接下来的奇数度节点用经过 floyd 预处理后的图矩阵，状态压缩 dp 下就

行了。floyd 的处理后，顺便可以判连通。