IOI2014 中国国家集训队第一次作业
ACM/ICPC World Finals 题目泛做

做题表格

姓名：梁泽宇

说明：
（1）试题编号、名称均为原题的真实编号、名称，Tsinsen 上的个别试题编号和名称有误；
（2）题目大意中的数据范围、数据组数均为 Tsinsen 上的版本，不指出多组数据的题目，均为一组数据；
（3）时空复杂度均为对每组数据的时间、空间复杂度。

试题编号	名称	题目大意	算法讨论	时空复杂度
1998 C	Lead or Gold	有 N 种由原料 A、B、C 混合得到的物质，给出它们各自的 A、B、C 的比例，求出是否可以用它们之间进行混合得到一个指定的比例 a:b:c。1<=N<=100。测试数据不超过 100 组。	经典问题。先考虑两种原料的情况，假设用(x, y)来表示一种已知的比例为 x:y 的混合物，则用(x1, y1)和(x2, y2)能得到比例 x0:y0 当且仅当(x0, y0)在线段(x1, y1)-(x2, y2)上，进一步，如果是 N 种已知混合物，可以很容易得到，x0:y0 能得到当且仅当(x0, y0)在这 N 种已知混合物比例点的凸包内部（含边界）。 然后考虑三种原料，一种很自然的思路是将三种化为两种，因为给出的 a:b:c 中必然有一个冗余的值。如果用除法，a:b:c 变成 1:(b/a):(c/a)，则 a=0 不好处理，所以用除法是不行的。正确的方法是将 a:b:c 变成(a/(a+b+c)):(b/(a+b+c)):(c/(a+b+c))，这样变化后的 a+b+c=1，只要知道 a、b 就可以得到 c 了。	时间复杂度： $O(NlogN)$ 空间复杂度： $O(N)$
1998 D	Page Selection by Keyword Matching	网页关键词匹配。给出一些网页和一些即时查询，各包含一些关键词，在目前的各个网页中查找这些关键词（不分大小写）并计算总权值，输出总权值最大的前 5 个（若总权值大于 0 的不到 5 个则全部输出）。每个网页和查询包含的关键词数目 N<=8，网页个数 M<=25，关键词长度 len<=20。	暴力模拟，按照题目所说的模拟即可。	时间复杂度： $O(N^2M \cdot len)$ 空间复杂度： $O(NMlen)$
1998 F	Polygon Intersections	给出平面上的两个简单多边形，求出它们的交集，要求对于交集的每个区域（必然都是简单多边形）按照顺时针输出所有的顶点，且不能出现多余顶点（连续三个顶点共线）。1<=多边形顶点数 N<=100，所有坐标的绝对值不超过 10^4。多组数据。	容易发现，简单多边形的交集一定也是若干个简单多边形，且它们的边界都是原来多边形边界的一部分，因此，只需要对于两个多边形，分别求出它的每条边和另一个多边形的交集中的每条线段，再用这些线段建立平面图，用求平面图区域的办法得到每个区域即可。因此，本题的关键问题在于如何求出一条线段和一个多边形的交集。使用传统方法需要考虑线段经过多边形顶点、甚至与多边形的边重合等许多特殊情况，极其麻烦。一种比较简单的做法是三角剖分——在平面上任取一个定点（不妨取原点），用它和多边形的每条边组成一个三角形，并且根据这条边的旋转方向给这个三角形赋一个权值（本题中顺时针为 1，逆时针为-1），然后依次求出该线段与每个三角形的交集（如果存在，必然是一个点或一条线	时间复杂度： $O(N^2 logN)$ 空间复杂度： $O(N^2)$

			段），如果交集的长度大于 0（是线段），就在交集中加入这条线段，其权值为该三角形的权值，然后再对这些线段作带权加法（先排序，然后扫描一遍，累加权值）即得到线段与整个多边形的交集。不过要特别注意一种特殊情况——当线段与多边形的某条边重合的时候，在处理这条边形成的三角形时，对于得到的交集线段，如果该三角形权值为 1，则线段权值为 1，若三角形权值为-1，则线段权值为 0，因为它不能去掉！！此外，为了防止出现多边形的某边经过原点导致剖分出的三角形退化为三点共线的情况，需要先将所有的点均平移到第一象限，输出时再减去平移量。 这种三角剖分的方法，在后面的某些题目中仍将使用。	
1998 G	Spatial Structures	存储一个 N×N（N 为 2 的整数次幂）的点阵，使用四分树，构造方法如下：根结点代表整个点阵，如果某个结点代表的区域内各个像素的颜色均相同（全白或全黑），则将其标记为白点或黑点，并不再细分，否则将其标记为灰点，并等分成左上、右上、左下、右下四个区域，分别用其编号为 1、2、3、4 的四个子结点代表。构造出的四分树可以用一个表示所有黑点位置的整数序列来表示，对每个黑点，将其到根的路径上的所有点的编号（即它是它父结点的第几个子结点）从高位到低位连接成一个五进制数，并将其转化为十进制数，再递增排序得到序列，如果根结点是黑点，则用只含一个 0 的序列表示，如果根结点是白点，用空序列表示。现在要求实现点阵与序列间的相互转换。 1<=N<=64，且 N 一定是 2 的整数次幂。有多组数据。	对于点阵转序列，先按照题目的要求递归构造出四分树，再构造出序列即可。对于序列转点阵，可跳过四分树这一步，直接对序列中的每个元素找出它代表的区域，再对此区域全部涂黑即可。注意判断整个点阵全白或全黑的特殊情况，以及输出序列时不要忘了输出格式中的“12 个一换行”。	时间复杂度： $O(N^2 \log_4 N)$ 空间复杂度： $O(N^2)$
1999 B	Bullet Hole	有一个大的正方体(0, 0, 0)-(d, d, d)，被分成 n^3 个(d/n)*(d/n)*(d/n)个小的正方体，这些小的正方体之间彼此隔开，每个小的正方体都充满水。现在在空间中的某个位置朝某个方向发射子弹，可能穿过这个大正方体，留下一些弹孔，求总共会漏出多少水。1<=n<=50，0<d<=100，所有坐标的绝对值不超过 100，zero 取 10^{-6}。多组数据。	对每个小正方体单独考虑，如果它没有被打穿，则它里面的水不会漏出，如果被打穿，则它里面 z 坐标大于最低弹孔位置部分的水都会漏出，因此问题转化为求一条射线（是射线！！）和一个正方体的交点（也可能是交线）。这里要对五个面（上表面的交点不计）分别求交点，还要处理该射线垂直于 x、y、z 轴、平行于 x、y、z 轴以及刚好位于某个面所在的平面内时的特殊情况，比较麻烦。	时间复杂度： $O(N^3)$ 空间复杂度： $O(1)$
1999 D	The Fortified Forest	给定平面上的 N 个点的位置，每个点有两个权值 w1 和 w2，现在要求取出一些点，使得它们的 w1 之和不小于剩下的点的凸包的总边长，且它们的 w2 之和尽可能小。当有多种使 w2 之和最小的方案时，选取取出	由于 N 的范围很小，暴搜即可（枚举选出的点，然后求剩下的点的凸包总长度，选取满足题目要求的最优方案）。在求凸包的时候 ，要注意处理剩下的点数<=2 以及剩下的点全部共线（需要总长度的两倍）的特殊情况。	时间复杂度： $O(2^N \cdot NlogN)$ 空间复杂度： $O(N)$

		的点数尽可能少的方案（保证唯一）。 1<=N<=15，0<=w1, w2<=10^4。多组数据。		
1999 G	The Letter Carrier's Rounds	模拟 MTA 之间的邮件通信过程，给出 MTA 之间发送的所有命令，输出其回复。 可能的命令有： HELO myname（发送发件 MTA 名字） MAIL FROM:<sender>（发送发件者名字） RCPT TO:<user>（给出一位收件者的名字） DATA（标志正文的传送开始。接下来若干行为邮件正文，以首字符为句号(.)的一行结束） QUIT（结束会话） 收件 MTA 对命令的答复返回码： 221（关闭连接，在 QUIT 之后） 250（行为正常，即 MAIL FROM 或 RCPT TO 命令所示名字存在，或邮件正文传输结束） 354（开始传送邮件，在 DATA 之后） 550（无此用户，即 RCPT TO 命令给出的收件者不存在） MTA 个数、用户个数、消息条数、消息正文占用的行数均不超过 10^5，MTA 和用户的名字由字母和数字组成，且长度不超过 15，消息正文每行长度不超过 72，输入中各个部分以首字符为星号(*)的一行隔开，这种行也表示输入的结束。所有出现顺序不明的地方，均按照输入中出现的顺序。	本题的主要难点在于，MTA 个数、用户个数、消息条数均可能达到 10^5 级别，因此在给出一个用户查找其是否存在，以及将用户按照 MTA 进行归类、排序的过程均不能使用暴力，而用 Trie 存储字符串又会导致 MLE。这时，我们可以想到 C++的一个有力工具——STL！！用 STL 的 set 和 map，可以很快解决这种复杂的存储和查询问题。 此外，本题的输入输出格式比较复杂，其中需要特别注意的是，分隔行的标准是“首字符为句号或星号”，可能后面还有字符，甚至可能后面还有空格！！所以不能用 scanf 进行输入（scanf(“\n”)在这里是没有用的，因为它只能在输入文件时的后面一个字符为\n 的时候过滤掉这个\n），gets()比较危险，不推荐使用，最好的办法是使用 while ((ch = getchar()) != '\n')来读入。	时间复杂度： $O(NlenlogN)$ 空间复杂度： $O(Nlen)$ 其中 N 为收入中的 N，不超过 10^5，len 为名字的最大长度 15。
2000 D	Gifts Large and Small	给出一个简单多边形，求出其旋转任意角度后的最小包围矩形面积的最小值和最大值。3<=多边形点数 N<=100，坐标的绝对值不超过 10^4。多组数据。	根据题意，最小包围矩形的面积等于$(x_{max}-x_{min})*(y_{max}-y_{min})$。一种很显然的方法是先枚举旋转后的四个极点（最大 X 坐标点、最小 X 坐标点、最大 Y 坐标点、最小 Y 坐标点）是什么，然后得出满足这个条件的旋转的角度范围，然后在这个角度范围内求解最小、最大面积。因此，先预处理出每个点成为最大 X 坐标点、最小 X 坐标点、最大 Y 坐标点、最小 Y 坐标点时，旋转角的范围是什么（比如求 i 成为最大 X 坐标点时旋转角的范围，只需要枚举其它所有的点 j，求出 i 的 X 坐标大于等于 j 时的旋转角范围，再求交集即可。容易发现任何情况下“i 的 X 坐标大于等于 j 时的旋转角范围大小”总是 π rad，故只需要求出“i 的 X 坐标等于 j 时的旋转角的值“即可。这个交集一定要么为空，要么也是一个角度区间，不过角度区间可能跨越 2π rad）。然后枚举旋转后的四个极点，得到旋转角范围（枚举了前 1～3 个，当交集为空的时候就可以不继续枚举，所以远远达不到 $O(N^4)$ 的枚举量）。接下来，问题转化为求	时间复杂度： $O(N^4)$ （其实远远达不到） 空间复杂度： $O(N^2)$ 或 $O(N)$

			$(A\cos\theta+B\sin\theta)(C\cos\theta+D\sin\theta)$ 当 θ 在一个区间内时的最小值与最大值的问题（其中 A、B、C、D 都是常数）。这个式子可先化为有关 2θ 的正余弦值的式子，再进一步化为 $$\frac{AC+BD}{2}+\frac{1}{2}\left[\sqrt{(AC-BD)^2+(AD+BC)^2}\cdot\sin(2\theta+\varphi)\right]$$ $$\sin\varphi=\frac{AC-BD}{\sqrt{(AC-BD)^2+(AD+BC)^2}},\cos\varphi=\frac{AD+BC}{\sqrt{(AC-BD)^2+(AD+BC)^2}}$$ 这样就很好求了。	
2000 E	Internet Bandwidth	给出一个无向图，每条边正反向容量相同，求出 S-T 最大流。1<=点数 N<=100，1<=边数 M<= C_2^N ，0<=边容量<=1000。	由于每条边正反向的容量相同，所以每条无向边可以拆成两条方向相反的有向边，然后就是求有向图 S-T 最大流了，用 Dinic。	时间复杂度： $O(N^2M)$ （Dinic） 空间复杂度： $O(N^2)$ 或 $O(M)$
2000 G	Queue and A	有 N 种请求，每种请求会等间隔地发出，有 M 个人可以来处理这些请求，每个人都有他可以处理的请求种类以及优先级，每种请求的处理都要耗掉一定的时间。任意时刻，若某个人空闲，他会从目前已经发出而尚未处理的请求中按照其优先级顺序找一个请求处理，如果没有他可以处理的请求则他会继续空闲。当两个人要同时处理同一个请求时，优先安排给上一次开始工作的时刻较早的那个，如果仍相同，优先安排给编号小的那个。给出每种请求发出的个数、第一个这种请求发出的时刻、这种请求发出的时间间隔、处理一个这种请求需要消耗的时间、每个人的编号、可以处理的请求种类及优先级列表，求出所有的请求全部处理完毕的时间。 1<=N<=20，1<=M<=20，1<=每种请求的个数 s<=1000，其它的数和结果不超过 2^{31}-1。多组数据。	按照题目的要求模拟。注意题目中共有两种事件，一是请求发出事件（发出一个新的请求），二是请求处理完毕事件（某个人处理了某个请求完毕，进入空闲状态），其中第一种事件的时刻预先可知道，第二种事件的时刻则需要在模拟过程中才能知道并加入事件。在模拟过程中，每到出现新事件的时刻结束后（这一时刻的所有事件发生后的一瞬间）要结算一次，即安排人处理目前的未处理请求，具体来说，先将所有在此时刻空闲的（包括刚刚开始空闲的）人按照上一次开始工作的时间递增，若相同按照编号递增排序，然后对于每个人，按照其优先级从目前未处理请求中找到一个请求处理，同时加入一个请求处理完毕事件（在他处理完这个请求的时刻）。特别注意，结算要发生在新事件开始前，而不能发生在新事件开始后，因为结算中加入的事件可能比原来的最新事件更早发生（不注意到这点就只有 20 分了……）	时间复杂度： $O(N^2Ms)$ 空间复杂度： $O(NM)$
2000 H	Stopper Stumper	给出一个三角形的三边长和三个瓶塞，判断这三个瓶塞是否能全部放入三角形。要求：不许上下叠放，但一个瓶塞可以搭在另一个瓶塞上（不增加高度），每个瓶塞都要与三角形的两边相切，且不能有两个瓶塞与三角形的同样的两边相切。瓶塞由两个底面大小不同（底面圆心相同）、高度相同的圆柱上下叠放组成。三个瓶塞的	必然是两个瓶塞正放、一个瓶塞倒放（一个正放两个倒放等价），因此总共有 18 种情况（枚举每个瓶塞与三角形的哪两条边相切，以及哪个瓶塞倒放）。接下来就是判断是否合法了，注意，只有当三个瓶塞都能放入三角形内（求出圆心，圆心在三角形内且大圆不越出三角形），且两个正放的瓶塞大圆不重叠（相交、内含或内切均为重叠）、正放的瓶塞大圆和倒放的瓶塞小圆不重叠、 正放的瓶塞小圆和倒放的瓶塞大圆不重叠时，才为合法。	时间复杂度： $O(1)$ 空间复杂度： $O(1)$

		高度都相同。多组数据。		
2001 B	Say Cheese	空间有若干个球（可能重叠），现在要从空间中的一点 S 走到另一点 T，在球内走不消耗时间（即可以瞬间从球内的一个点走到另一个点），在球外走的速度是一定值。求从 S 到 T 的最短时间。0<=球的个数 N<=100，所有坐标的绝对值和球的半径不超过 10^4。数据不超过 10 组。	先预处理求出从球 i 走到球 j（不考虑其它球）的最短时间，为 $\frac{min\{dist(O_i,O_j)-R_i-R_j,0\}}{v0}$，其中 v0 为速度（0.1），以及从 S 和 T 走到每个球的最短时间。然后将每个球作为一个点，S 和 T 作为起点和终点，构造无向图，用 Dijkstra 求 S-T 最短路即可。注意 S 和 T 本身位于球内的特殊情况。	时间复杂度：$O(N^2)$ 空间复杂度：$O(N^2)$
2001 E	The Geoduck GUI	N*M 的网格中有 K 个动点，每个动点有一个移动向量（X、Y 坐标均非 0），一开始（t=1 时）每个动点位于其移动向量的对角格子，每次横向或纵向移动一格，移动到的格子为从该动点的初始格子的中心出发，方向向量为其移动向量的射线下一个穿过的格子，如果穿过角，就先横向再纵向移动，移出边界后会回到同行/列的另一侧。当某个动点移动到某个已被任意一个动点经过的格子时停止，两个动点同时移动到同一个格子或同时试图移向对方的格子时，两者同时停止。现在要从 K 个动点中选出两个，使得经过的格子尽可能多，并且在此基础上用时尽可能少。1<=N, M<=50，2<=K<=10。多组数据。	枚举选哪两个，然后按照题目所说的模拟即可。为了加快速度，可以先预处理出每个动点的移动轨迹。 注意两点，一是预处理移动轨迹时，移动射线可能会越界，此时不能将其回到另一侧；二是模拟两个动点移动的时候，终止的条件是不再有新的格子被占据，而不是两者均停止。	时间复杂度：$O(NMK^2)$ 空间复杂度：$O(NMK)$
2001 G	Fixed Partition Memory Management	有 N 个程序和 M 个内存分区，一个程序在每个分区内运行需要的时间可能不同（也可能某个程序不能在某个分区中运行），一个分区在任意时刻最多只能运行一个程序。给出一种将程序分配到内存分区的编号及运行顺序的方案，使得所有程序的平均等待时间最小。1<=N<=50，1<=M<=10。	本题即为 NOI2012 delicacy 的原题（delicacy 增大了数据范围，使得只有使用动态加点才能过）。 先考虑暴力做法，建出二分图，X 方点表示程序，Y 方点表示内存分区和该分区内运行的第几个程序的二元组，X、Y 方点之间的边容量为 1，费用为某程序在某分区内作为第几个运行的程序，对总等待时间贡献的效果值，设置源 S 汇 T，S 向所有 X 方点以及所有 Y 方点向 T 都连一条容量为 1 费用为 0 的边，求 S-T 最小费用最大流即可。这种方法建出的图总点数为 $O(NM)$，总边数更是达到了 $O(N^2M)$。 考虑对这种算法进行优化。容易发现，对于某个分区来说，如果其“第 i 个运行的程序”这个 Y 方点没有使用，则其“第 j（j>i）个运行的程序”这个 Y 方点显然也不能使用，因此，可以在增广过程中记录下每个分区已经使用了多少个 Y 方点，然后动态加入新的点。由于有且只有 N 个 Y 方点被使用，所以在整个过程中图的点数都为 $O(M+N)$ 级别，边数都为 $O(N(M+N))$ 级别。	时间复杂度：$O(N^2(M+N))$（使用动态加点） 空间复杂度：$O(NM)$
2001 H	Professor Monotonic’s Networks	给出一个比较网络，由 M 个比较器组成，有 N 个输入和 N 个输出，一个比较器可以将两个输入值进行排序。问这个比较网络	将比较器作为点，如果比较器 i 的输出作为比较器 j 的输入，则连边<i, j>。在这个图上求最长路径的长度（由于边权都是 1，用 BFS 或者拓扑排序后递推一遍即可）即为第二问的结果。对于第一问，	时间复杂度：$O(S(M+N))$（其中 S 为随机次

		是不是排序网络（不管输入是什么，输出的结果都有序），以及它的运行时间。1<=N<=12，0<=M<=150。	一种比较简单的方法是随机——随机许多输入，看相应的输出是否有序。	数） 空间复杂度：$O(M+N)$
2002 A	Balloons in a Box	给出一个长方体，有 N 个点，每次可以选定一个在该长方体内且不在任何一个气球内的点上放一个气球，然后让其以该点为球心膨胀，直到碰到其它气球或长方体边界为止。可以以任意顺序选定这些点，使得最后所有气球占据的总体积最大。1<=N<=6。	由于数据范围非常小，暴搜即可。那些不在长方体内的点可以预先排除。	时间复杂度：$O(N\cdot N!)$ 空间复杂度：$O(N)$
2002 B	Undecodable Codes	给出 N 个 01 串，求出最短的 01 串满足它能用这 N 个 01 串以至少两种不同方式拼成，保证有解，如果有多解，输出字典序最小的。1<=N<=20，1<=每个 01 串长度<=20。不超过 100 组数据。	设 F[i1][j1][i2][j2]表示两种方式分别匹配到第 i1 个串的第 j1 个字符、第 i2 个串的第 j2 个字符的最短匹配长度。很明显，这可以 BFS 求出（一开始满足 i1≠i2、j1=j2=0，且第 i1、j1 个串的首字符相同的状态值为 1，其它状态值为 INF，然后每次用一个状态去推出其它的状态，BFS）。当两种匹配方式同时到达对应串的末尾时，表示已找到了最优解。 然后，本题要求多解时输出字典序最小的，这有两种处理方法。一是一开始将所有 01 串按照字典序递增排序，然后在后面 BFS 的过程中，若 i1 和 i2 两个串一个到末尾而另一个没到，则枚举新的串时按照排序后的编号递增枚举，这样就可以保证找到的第一个解一定是字典序最小的。二是通用方法——标号法，即将字典序与数字（标号）一一对应，满足字典序越小的标号也越小，字典序相同的标号也相同即可。由于本题 BFS 过程中值总是加 1，所以可以划分阶段，故可以使用静态（离线）标号法。初始状态（第一个字符）中，值为 0 的标号为 0，值为 1 的标号为 1，然后每次用一个阶段递推下一个阶段时，先将本阶段的各个状态按照标号（即字典序）递增排序，然后递推时，新的状态的标号为按照先前趋状态标号、后本阶段值（0 或 1）递增排序后的结果。这样直到找到解为止，即为字典序最小解。	时间复杂度：$O(N^2 len^2 logN)$ 空间复杂度：$O(N^2 len^2)$ 其中 len 为 01 串最大长度。
2002 G	Partitions	一个矩形的划分指的是在其内部加入若干条（可以是 0 条）水平和竖直线段，使得它们分成的每个部分都是矩形，且不存在多余的线段。对于一个矩形，如果其存在两个划分 A 和 B，满足 A 的线段集合真包含于 B 的线段集合，则称 A 比 B 粗糙，B 比 A 精细。现给出同一个矩形的两种不同的划分，满足它们中的任何一个都不比另一个粗糙或精细，求出它们的“上确界”（比两者都粗糙的划分中最精细的）和“下确界”（ 比两者都精细的划分中最粗糙	显然这两个划分的线段集合的并就是它们的下确界的线段集合，但这两个划分的线段集合的交却不一定是它们的上确界的线段集合，因为其中可能出现多余的线段，也可能出现某个连通块不是矩形。不妨设 S 为这两个划分的线段集合的交，则可以得到，它们的上确界的线段集合是 S 的子集。因此，剩下的问题就是删去 S 中的一些线段从而得到上确界。 先将 S 画在矩形中，称长度等于单位长度的线段为“单位线段“，由单位线段（包括边界上的单位线段）组成的完整的矩形称为”生成矩形“，选出这些生成矩形中的尽可能多个，满足它们能覆盖整个矩形且两两不相交，然后删去所有选出的生成矩形内部的边即可。容易发现，对于矩形中的任意一个格子，都存在且只存在一个最小的生	时间复杂度：$O(N^2 M^2 (N+M))$ 空间复杂度：$O(N^2 M^2)$

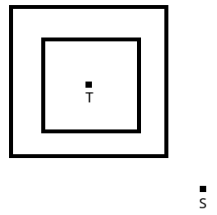
		的）。1<=矩形的长度 N、宽度 M<=20。不超过 6 组数据。	成矩形（其长度和宽度都达到最小）包含这个格子。所以，将所有的生成矩形求出后，按照面积递增的顺序扫描，如果某个生成矩形满足其内部还有至少一个格子未被选出的生成矩形覆盖，则选出它，并且删去所有它包含的生成矩形，扫描完后就得到了应该选出的生成矩形集合。	
2003 A	Building Bridges	给出一个 N×M 的网格，表示城市示意图，其中的某些格子被建筑占用，若干相邻（八连通）的占用格属于同一个建筑。现在要在其中建立道路连接尽可能多的建筑，道路必须是直线，只能沿着网格边界建路，且一条道路必须连接两个不同的建筑（中间不能经过其它建筑）。求出建立道路数最少的方案，如果有多解，要求道路的总长度尽可能小。1<=N, M<=50。多组数据。	先求出每个格子属于哪个建筑，以及所有能建的道路（扫描一次即可，最多只有 2NM 条边）。然后，以建筑为点、道路为边，用 Kruskal 求最小生成森林，即为结果。注意题目中所说的各种特殊情况。	时间复杂度：$O(NMlogN)$ 空间复杂度：$O(NM)$
2003 C	Riding the Bus	有一种递归定义的 SZ 曲线（具体定义方法见原题），现在要求出在 N 阶 SZ 曲线中，从一个点 S 走到另一个点 T（先从 S 直线走到最接近的 SZ 曲线连接点上，再沿 SZ 曲线走到最接近 T 的连接点，再直线走到 T）的总长度。1<=N<=8。多组数据。	先求出离 S 和 T 最近的连接点 S'和 T'，这个好求，求出 N 阶 SZ 曲线的单位长度 len，再找到离 S 和 T 最近的 len 倍数点即可。 然后求 N 阶 SZ 曲线上 S'到 T'的距离，这就是求出 S'和 T'分别是 N 阶 SZ 曲线的第几个连接点。这个显然要递归来求，从 1 阶开始不断展开。注意，曲线有 S 型和 Z 型，两种曲线又各有从左开始和从右开始两种，特别需要注意的是每一阶的单位长度都是不一样的。	时间复杂度：$O(N^2)$ 空间复杂度：$O(N)$
2003 E	Covering Whole Holes	给出两个只有 90 度或者 270 度的角、所有边都与坐标轴重合的简单多边形 A 和 B，判断 A 是否能完全嵌入 B 中（可以在边界上重合，但 A 不能超出 B 的边界，A 和 B 可以平移但不能旋转、翻转）。4<=A 的边数 N，B 的边数 M<=50。多组数据。	注意本题中 A 和 B 的特点是只有水平和竖直边，且水平和竖直边交替出现，所以，可以采用矩形剖分的办法。以下假设 A 和 B 的结点按照逆时针排列，且 A 和 B 均完全位于第一象限（否则将其平移到第一象限即可）。 第一步，枚举 A 在 B 中的放入位置，注意只需满足 A 的一条水平边与 B 的一条水平边重合，且 A 的一条竖直边与 B 的一条竖直边重合即可。枚举时要特别注意的是，A 的那条水平边是可以长于 B 的那条水平边的，竖直边也一样。 第二步，判断平移后的固定的 A 是否完全位于 B 内。判断凹多边形位于另一个凹多边形内部，用传统方法很麻烦，使用剖分法则简单许多，本题可以采用矩形剖分（当然前面说过的三角剖分也可以，不过麻烦一些）。预先将 B 的所有水平边投影到 X 轴上，所有竖直边投影到 Y 轴上，得到若干个有向面积的矩形（由于都在第一象限，水平边向右为正向左为负，竖直边向上为正向下为负），然后，A 完全位于 B 内等价于 A 的每条边都完全位于 B 内，而判断 A 的一条边是否完全位于 B 内，只要求出这条边在 B 剖分后的每个有向面积矩形内的部分的长度（有向长度，注意水平边只要判断水平边引出的矩形，竖直边只要判断竖直边引出的矩形），看加起来是否等于这条边的总长度即可，注意特殊情况：如果该边刚好在矩形的上/右边界上，则当该矩形面积为正时，应算上，面积为负时，应不算。	时间复杂度：$O(N^3M^3)$ 空间复杂度：$O(N+M)$ 这里 N 和 M 均为原题中的一半。

2003 G	A Linking Loader	给出若干个有序链接的编译模块，每个编译模块内有若干个定义标识符（并申请内存）的操作（D 操作）、若干个声明标识符的操作（E 操作），以及 若干个批量赋值（其中可能引用已在本模块中声明的标识符的内存地址）操作（C 操作）。所有标识符都是全局的，并且 E 操作中声明的标识符可能在当前模块后面的模块中定义，也可能未定义或被多次定义（未定义时，内存地址取$(0000)_{16}$，多次定义时取第一次定义的内存地址）。模拟这个整合模块的过程，并且输出其整合之后，内存中的所有标识符的地址和所有存储的值的十六位校验和（在原题中定义）。D 操作不超过 100 个，内存地址不超过$(FFFF)_{16}$。多组数据。	可以发现标识符的内存分配与目前内存的使用状态无关，即这里的“标识符”只起到指针的作用，不额外占用内存，只有 C 操作会占用内存，且占用的一定是从起始位置$(0100)_{16}$开始的若干连续的内存。因此，解决本题需要两步，第一步，忽略 E 操作，根据 D 和 C 操作求出每个符号的内存（地址），第二步，忽略 D 操作，根据 E 和 C 操作求出内存中的每个位置存储的值（数字）。 需要注意的是， 题目中所有的数字都是十六进制的，包括 C 操作中的 n。	时间复杂度： $O(N+M)$ 空间复杂度： $O(N+M)$ 其中 N 为操作的总数，M 为占用内存的总数。
2003 I	The Solar System	给出太阳系中一个行星轨道的半长轴长度、半短轴长度和环绕周期，以及另一个行星轨道的半长轴和半短轴长度，用开普勒定律求出“另一个行星”在 T 单位时间后运行到哪里。假设这“另一个行星”的轨道（椭圆）沿坐标轴对称，焦点位于 X 轴上，太阳位于 X 坐标为正的焦点处，0 时刻该行星位于 X 坐标最大处。所有输入数据都在 int 范围内，不超过 100 组数据。	先根据开普勒第三定律求出待求行星的环绕周期，然后 T 对这个环绕周期取模。 然后，根据开普勒第二定律，连接太阳所在焦点和这个行星之间的线段在相等的时间内扫过相等的面积，所以，可以求出在 0～T 时刻（取模后的），该线段扫过的面积（椭圆面积为 abπ，除以周期再乘以 T 即为扫过部分的面积）。接下来，具体坐标不好直接求出，可以二分（先求出这个结果在第几象限，再在该象限内二分 X 坐标），然后求面积（用简单定积分即可求出），这样直到精度足够为止（需要精确到 10^{-10}，否则无法过），注意，整个过程中所有的浮点数都要用 long double！！！	时间复杂度： $O(logMAXS)$ (MAXS 为二分范围） 空间复杂度： $O(1)$
2004 B	Heliport	给出一个仅由水平和竖直线段组成的简单多边形，求出其最大内切圆的大小。4<=多边形边数 N<=20，边长不超过 50。多组数据。	本题是基于一个图形在另一个图形中的嵌入的问题。一般对于这种题的基本思路就是，找到该嵌入图形不断平移后必然出现的特殊情形（碰到被嵌入图形的某个顶点或某条边，等等），然后进行枚举、判断是否可行的操作。本题是圆在多边形内嵌入，先二分圆的半径 r，容易发现半径为 r 的圆如果能嵌入多边形，经过平移后必然出现以下三种情况之一：（1）碰到多边形的两个顶点；（2）碰到多边形的一个顶点并且和多边形的一条边相切；（3）和多边形的两条边（一条水平一条竖直）相切。枚举这三种情况，可确定该圆的圆心，然后判断是否合法（圆心在多边形内，且圆心到多边形的每个顶点的距离都不小于 r，且圆心到多边形的每条满足 x 坐标位于两端点之间的水平边和 y 坐标位于两端点之间的竖直边的距离都不小于 r）即可。这里需要判断圆心是否在多边形内，方法除了传统的引射线法外，还有一种就是剖分法（三角剖分和矩形剖分均可，但要预先判断是否在边界上）。	时间复杂度： $O(N^3 logMAXS)$ (MAXS 为二分范围） 空间复杂度： $O(N^2)$（矩形剖分）或 $O(N)$（射线法或三角剖分）
2004 C	Image Is Everything	有一个由单位立方体组成的立体图形（立方体可能不连通），每个单位立方体的六	本题用确定性算法很困难，因为三维数组是非线性的。对于这类因非线性导致无序的问题，一种常见的方法是松弛算法。对于本题，	时间复杂度：

		个面上都染上相同的颜色。现给出整个图形在六个面上的视图，求出它最多可能由多少个立方体组成？保证有解。1<=立体图形的边长 N<=10，不超过 5000 组数据。	可以先假设这个图形充满了立方体（N×N×N 个），然后，将那些多余的立方体不断除去，直到剩下的图形满足视图要求为止（得到的显然一定是最优方案）。一个立方体是多余的，当且仅当从某个面上它的位置显示是空的，或者从各个面上看到它的颜色不完全相同。由于本题数据组数较多，为了加快速度，需要对各个面上的各个格子建立 Dancing Link，这样可以在 O(1)的时间内找到每个面上的第一个（即能看见的）立方体，以及在 O(1)的时间内删除一个立方体。	$O(N^3)$（期望） 空间复杂度：$O(N^3)$
2004 D	Insecure in Prague	给出一个由大写字母组成的字符串，用以下方法对其进行加密：首先确定 N（不小于字符串长度的两倍）、s、t、i、j（1<=s, t, i, j<=字符串长度，且 i<j）的值，然后构造 N 个空位，将字符串的首字符填入第 s 个空位，接着向右每过 i 个空位填上字符串的下一个字符，如果越过最右边的空位则回到最左边，直到填上整个字符串为止；然后，再将字符串的首字符填入第 t 个位置或其右边第一个空位，接着向右每过 j 个空位填上字符串的下一个字符，如果越过最右边的空位则回到最左边，直到填上整个字符串为止，最后，对剩下的空位随机填入大写字母。问，仅给出加密后的字符串（由所有空位上的字符顺次连接而成），则原字符串最长能是什么？最长的原字符串是否唯一？2<=加密后的字符串的长度 N<=40，且满足至少有一个大写字母在该串中出现不止一次。多组数据。	先枚举 s 和 i 的值，这样就可以得出原字符串（如果足够长的话）的每个字符（中间加入剪枝：如果原字符串中的某个字符出现的次数超过了加密后字符串中该字符出现次数的一半，则剪枝），然后，枚举长度，就可以得到原字符串，再通过枚举 t（满足目前第 t 个位置是空位且加密后字符串的第 t 个字符与枚举的原字符串的首字符相同）和 j（>i）的值，就可以验证这个枚举的原字符串是否合法。注意，整个过程都要使用 Dancing Link 来优化，否则 TLE。	时间复杂度：$O(N^6)$（其实由于剪枝的作用，枚举量远远达不到这么多） 空间复杂度：$O(N)$
2004 F	Merging Maps	给出 N 张规模相同的矩形照片，编号 1～N，照片中的某些部分可以明确辨认，用大写字母表示，另一些部分不能明确辨认，用-表示。两张照片可以合并，当且仅当可以将其中的某些部分重叠在一起，满足重叠的部分的大写字母对应（即如果两张照片这个地方都是大写字母，则必须相同，-可以与任何字符对应）。两张照片的某种合并方案的得分为其重合的大写字母个数。现在要将 N 张照片进行尽可能多次合并，每次取可以合并的得分最高的方案，如果有多个得分最高的方案，取（编号较小的照片编号，编号较大的照片编号，行偏移量，列偏移量）的字典序最小的方案，合并后的照片赋予之前最大的编号加	按照题目要求进行模拟，每次枚举所有的合并方案，并判断是否合法，选取最优合并方案即可。在实现过程中需要注意以下几点： （1）由于合并后的照片可能不是完整的矩形，用二维数组来存储照片并不好，应该使用“坐标+值，即(x, y, v)”的方法来存储照片，不过，为了方便枚举，对每张照片要记录下其 x 跨度 xs 与 y 跨度 ys（最好在合并之后，将 xmin 和 ymin 归 0，这样 xs=xmax+1，ys=ymax+1）； （2） 合并两张照片的时候，如果同一个位置既有-又有字母，则应该取字母，这样-和未被覆盖到的位置就要区分开来； （3） 在验证两张照片是否可以以某种(r, c)合并的时候，最好枚举实际大小（实际占用的格子数）小的那个，将大的那个预先存储在二维数组里，这样可以加快速度，防止 TLE。	时间复杂度：$O(N^3 N_R^2 N_C^2)$（实际上达不到） 空间复杂度：$O(N^2 N_R N_C)$

		1 作为编号。重复进行以上操作直到剩下的任意两张照片都不能合并为止，输出最后的所有照片。1<=N<=10，1<=初始照片的长度 N_R 和宽度 N_C<=10，不超过 10 组数据。		
2004 G	Navigation	平面上有 N 个动点，每个动点都在朝着一个固定的方向以 100m/s 的速度匀速运动。每个动点在运动了一段时间后，发出信号，信号以 350m/s 的速度向各个方向传播。已知这些信号在同一时刻到达同一点，给出所有动点在 0 时刻的坐标、运动方向、信号发出的时刻，以及所有信号汇聚的时刻，求出信号的汇聚点，或者报告可能的汇聚点有多个或不存在。当两个点之间的距离不超过 0.1m 时，应当被视为同一点。1<=N<=10，其它所有数字绝对值不超过 10^4，不超过 5 组数据。	本题其实就是求 N 个圆的交点问题。这一问题有很多较为高效的算法，比如随机增量（可以在期望 $O(N)$ 时间内解决）。但是，本题却不能使用这些高效做法，因为本题的精度问题——距离的 zero 竟然是 0.1！这些高效做法几乎都会因为精度误差而无法得到正确结果。 所以，本题只能采用最暴力的办法了——任取两个相交（或相切）的圆求出交点（如果任意两个圆都相离，则所求汇聚点显然不存在），然后看这个交点（如果取出的两个圆相交的话，有两个交点）是否在其它所有的圆上（注意距离 0.1 的精度）即可。注意，当 N=1 时，该圆的半径不超过 0.05 时，有唯一解，否则有多解。	时间复杂度： $O(N^2)$ 空间复杂度： $O(N)$
2004 I	Suspense!	在两幢楼之间建造吊桥，吊桥由一条抛物线和一条水平直线组成，满足水平直线比抛物线的最低点低 1m。吊桥还需要满足以下条件： （1）抛物线的两端点为两幢楼上的指定的点； （2）水平直线至少比地面高 1m，至少比抛物线任意一个端点低 2m； （3）楼上的某些地方有猫，另一些地方有鸟，要求这条水平直线不能提供猫抓到鸟的道路。具体来说，要么在直线的下方 0.5m 到上方 3m（包括下方 0.5m 和上方 3m）范围内没有猫，要么在直线的下方 3m 到上方 0.5m（包括下方 3m 和上方 0.5m）范围内没有鸟。 给出两幢楼之间的距离 d、两幢楼上指定的点（作为抛物线的端点）的高度，以及两幢楼上所有猫和鸟的位置高度，求出满足以上条件的最长的抛物线长度。d<=25，1<=每幢楼上猫和鸟的总个数 N<=25，最多有 10^5 组测试数据。	先根据题意及已知条件求出抛物线最低点的可能高度（由若干段区间组成），容易证明，当抛物线的长度达到最长时，最低点的高度一定达到最低。因此，取最低的高度，计算此时抛物线的总长度即为答案。	时间复杂度： $O(NlogN)$ 空间复杂度： $O(N)$
2004 J	Air Traffic Control	平面上有 N 个点，现在有 M 个控制器，每个控制器的控制范围是一个圆，该圆内（不含边界）的点都被该控制器控制，而该圆边界上的点可能被控制也可能不被控	由题意得，任意一个控制器的边界上都至少有一个给定点。所以，对于每个控制器，枚举这个在其边界上的给定点（满足它不和已给出的两个边界上的点共线），这样就可以唯一确定这个圆，然后，按照给定的优先级确定被该圆控制的各个点（注意要排除两种不合	时间复杂度： $O(MN^2)$ 空间复杂度：

		制（但不能都不被控制）。在有多个点可能被控制时，优先考虑 Y 坐标大的，Y 坐标相同时优先考虑 X 坐标大的。现给出 N 个点的坐标、各个控制器控制范围上的两个点（不与 N 个点中的点重合）坐标、以及各个控制器控制的点数，求出被 0、1、……、M 个控制器控制的点各有多少个。1<=N<=100，1<=M<=10，每个控制器至少控制一个点。多组数据。	法情况，一是这个圆内部的给定点数不小于该控制器控制的点数，二是该圆内部和边界上的总给定点数小于该控制器控制的点数），得出按照优先级的最优解即可。如果某个控制器的各种可能范围均不合法，则问题无解。	$O(N+M)$
2005 A	Eyeball Benders	给出两个图 A 和 B，都由若干水平和竖直的线段组成，求出 A 是不是 B 中某部分的放大图（要求 X 和 Y 等比例放大，且 A 中的至少一个端点同时也是 B 中这个部分的线段端点），1<=A 和 B 中的线段条数 N 和 M<=50，-1000<=坐标<=1000。保证 A 和 B 内至少有一个线段交点。	由于 A 和 B 内都至少有一个交点，所以找到 A 内的任意一个交点(x0, y0)，它一定对应的是 B 中的某个交点，枚举这个与它对应的 B 中交点(x1, y1)，接下来就是确定放大比例。若在 A 中，存在至少一个线段端点的 X 坐标不等于 x0 且 Y 坐标不等于 y0，那么这个端点就可以作为指示点，具体来说，设(x0, y0)到这个点的方向为 P，则在 B 中从(x1, y1)开始，向 P 方向作射线，看这条射线与哪些线段相交（交点显然不会超过 N 个），枚举交点即可确定放大比例，然后验证是否对上即可。若在 A 中不存在 X 坐标不等于 x0 且 Y 坐标不等于 y0 的线段端点，也就是所有的线段都在 x=x0 或 y=y0 上，则看 A 中是否有不在边界上的线段端点，若有，还是可以确定比例（枚举 B 中对应的端点即可），若无，则 A 中只有两条线段 x=x0 和 y=y0，直接看 B 中的(x1, y1)处可不可以继续向上下左右延伸即可，注意要延伸到至少一个线段端点，因为题目要求“A 中的至少一个端点同时也是 B 中这个部分的线段端点”。	时间复杂度：$O(N^3)$ 空间复杂度：$O(N)$
2005 B	Simplified GSM Network	给出平面上的 N_1 个关键点，每个关键点控制的范围为平面上所有满足到该关键点的距离在到所有关键点的距离中最近的点。然后，再给出一个 N_2 个点、M 条边的无向图，点都是平面上的点，回答若干个询问，每次询问对于两个图中的点 S_i 和 T_i，在图中从 S_i 走到 T_i 至少需要经过多少个关键点的控制范围。1<=N_1, N_2<=50，1<=M<=2500，询问个数不超过 10 个，且不会出现各种特殊情况（两个关键点重合、两个图中的点重合、图中的点在关键点控制范围的边界上、图中的边与控制范围的边界重合、图中的边上有是三个或三个以上的关键点控制范围的边界的点等）。多组数据。	一开始求出每个关键点的控制范围（一定是一个凸多边形）。这个问题是 Voronoi 图，可以用专门的求 V 图的算法在 $O(NlogN)$ 时间内求出，当然，由于本题数据范围小，也可以用暴力半平面交的办法得到。 接下来，需要求出每条边的边权，可以转化为求一条线段和一个凸多边形是否存在交集的问题，只需要判断线段的两端点是否在多边形内，以及线段和多边形的边是否存在交点即可。 最后，求完边权后，做一次 Floyd 即可得到所有的最短路长度。	时间复杂度：$O(N_1^2(logN_1+M)+N$ 空间复杂度：$O(N_1^2+N_2^2)$
2005 C	The Traveling Judges Problem	给出一个带权无向图，它有一个关键点和若干个标记点，现在要求出它的一棵生成树，使得该生成树上所有位于关键点到至少一个标记点的路径上的边的权和最小。	暴力枚举选出的点集合（不在这个集合里的点视为不存在，关键点和所有标记点必选），然后求出这些点的导出子图的最小生成树（用 Prim 求，可以顺便求出所有点的父结点），如果最小生成树存在且解更优，则保存这个解（记录所有点的父结点，方便输出方	时间复杂度：$O(2^{N-1-N_j}N^2)$ 空间复杂度：

		点数 N<=20，标记点数 N_j<=10，1<=边权<=100，要求输出方案。	案）。	$O(N^2)$
2005 D	cNteSahruPfefrlefe	将一个排列(0, 1, …, 51)每次乘上置换(27, 1, 28, 2, …, 52, 26)，每次乘置换之后可能会交换两个相邻的位置（也可能不交换），给出这样操作不超过 10 次（不知多少次）的结果，求出操作次数以及那些进行过的交换的情况。多组数据。	首先注意到每次交换最多只能让两个位置出错，因此 K 次交换最多能让 2K 个位置出错，由于操作次数不超过 10，每次操作完后最多只能交换一次，所以最多有 20 个位置出错。预处理出 1～10 次操作（不交换）后的结果，然后看输入数据和其中的哪个满足 32 个以上的位置对上，即可得到操作次数。 然后搜索交换情况。注意一个性质（可以通过实践验证），就是进行 1～10 次不交换的操作，不可能将两个原先处于相邻位置的数重新弄到相邻位置。这样以来，每次枚举交换哪两个相邻位置的时候，如果这两个位置上的数都是对的，则这里肯定不能交换，否则（一对一错或两个都是错的），可以交换。搜索的时候，可以人为改变搜索顺序，先搜两个都是错的的情况，再搜一对一错的情况，这样可以更快找到较优解。此外可以加入一个剪枝：如果目前还剩最多 K 次交换，而错误的位置个数超过 2K，则无解，剪枝。	时间复杂度：由搜索量而定 空间复杂度：$O(NM)$ 其中 N=52，M=10。
2005 F	Crossing Streets	给出平面上的 N 条水平和竖直直线，以及一个起点和一个终点，求出从起点到终点的一条曲线至少要穿过多少条直线。 1<=N<=500，所有的坐标绝对值不超过 2×10^9。多组数据。	本题当然可以用平面图做，但由于图不连通，且各个面较难求出，不推荐用平面图，最好的方法是先离散化成网格，然后在网格中求 S-T 最短路。 问题是，该怎么建立这个网格？先看怎么取 X、Y 坐标进行离散化。如果只取题目中线段本身的 X、Y 坐标，因为遇到以下形式的数据会出错（离散化后的网格图中，根本无法走进那个回字形）：  因此正确的方法是，找线段本身的 X、Y 坐标的相邻两个坐标的中点（也可以将上述两种坐标一起取出建网格图，但会 TLE），然后建出的网格图中，跨越原来线段的边权为 1，没跨越的边权为 0，至于 S 和 T，取其所在格子的任意一个顶点都行。 此外还要注意的是，本题在求最短路的时候，由于点数达到 10^6 级别，直接用 SPFA 或 Dijk+Heap 求都会 TLE。注意到边权只有 0 和 1 两种，因此可以先求出由 0 边组成的各个连通块，然后再以连通块为整体求最短路（不必缩点，只要在 SPFA 的时候，更新一个点的最短路后就更新其连通块中所有点的最短路并入队即可）。	时间复杂度：$O(N^2)$ 空间复杂度：$O(N^2)$
2005 G	Tiling the Plane	给出一个只含水平边和竖直边的简单多边形，求出它是否能用四点或六点覆盖的方式铺满平面。1<=多边形周长 N<=50。多组数据。	由于判定的方法已经在题目中给出，所以只要枚举这些分界点就行了。对于四点覆盖，枚举 A、B 两个分界点，C 的位置可以通过周长算出（A 到 C 刚好是周长的一半），D 的位置可以通过 A 到 B 的长度算出。对于六点覆盖，枚举 A、B、C 三个分界点，其它的点类似算出。枚举后验证路径是否重合即可。	时间复杂度：$O(N^3)$ 空间复杂度：$O(N)$
2006 C	Ars Longa	一个 3D 物理模型，为一个无向图，点为 1kg 的球，（可视为质点），边为杆（可以承受任意大小的拉力）。放在地面上（z	设每条边上的拉力大小为 $x_1, x_2...x_M$，将其作为 M 个未知数，对于每个不在地面上的点，假设它静止（在其所有关联的边的拉力和自身重力作用下平衡），可列出 3 个方程（x=0、y=0、z=0。重力大	时间复杂度：$O(N^2M)$

		坐标为 0）的点固定，其它点可以绕边旋转。求这个模型是否能在重力作用下保持静止，若能，求它是否稳定（即不会因小幅扰动而移动）。1<=点数 N，边数 M<=100，不超过 10 组数据。	小可视为 1），得到一个方程组。该方程组的解数即为结果：无解为 NON-STATIC，唯一解为 STABLE，多解（无穷多组解）为 UNSTABLE，高斯消元即可判断。注意本题精度，zero 最多取到 10^{-10}。	空间复杂度： $O(NM)$
2006 E	Bit Compressor	将一个 01 串压缩，压缩方法为 K 个连续的 1 变为 K 的二进制表示（K=1 或 2 时，将 K 个 1 原样带下来，不压缩），0 直接带下来。给出压缩后的串以及原串的长度、原串中含有 1 的个数，求出符合条件的原串是否存在，若存在，是否唯一。原串长度<=2^{17}，压缩后的串的长度<=40。	由于 0 是直接带下来的，可以从 0 入手。由题意可以得到原串中 0 的个数 n0，设压缩后的串的 0 个数为 m0，若 m0<n0，则无解，否则问题转化为在压缩后的串的 m0 个 0 中选出 n0 个作为原串中的 0，枚举这些 0 的位置，然后，压缩后的串中相邻两个 0 之间的部分（如果有的话，为了简化处理，可以在压缩后的串的开头和结尾各加上一个 0）必然是原串中对应的两个 0 之间的 1 的个数的二进制表示，如果这个二进制数不合法（有前导 0、等于 2、超过原串中的 1 的个数加 1 等等）则该枚举方案不合法（这个二进制数的准确值可以预处理求出，没必要每次都算），否则累加进 1 的个数，中间还可以加入一些可行性剪枝加快速度。特别要注意的是，如果压缩后的串的两个 0 之间出现"11"，则它既可能表示两个 1，也可能表示三个 1，遇到这里要分两种情况考虑。	时间复杂度： 由搜索量而定 空间复杂度 $O(M^2)$ 其中 M 为压缩后的串的长度。
2006 F	Building a Clock	用转轴和一些齿轮构造一个钟。有且只有一个转轴，称为主动轴，其转速已知，其它转轴的转速由它与别的转轴的连接情况决定。共有 N 个齿轮，每个齿轮最多只能用一次。齿轮套在转轴上，一个转轴上最多能套三个齿轮，且各个齿轮的转速相同。一个齿数为 T1 的齿轮与一个齿数为 T2 的齿轮咬合，则两者的转速之比为-(T2/T1)。转速的正负号表示旋转方向，正数表示顺时针，负数表示逆时针。求出一种方案，使得：（1）至少存在一个转速为每小时一圈且顺时针旋转的转轴，作为分针轴；（2）至少存在一个转速为 12 小时一圈且顺时针旋转的转轴，作为时针轴。如果存在多个方案，选取使用转轴数最少的，若还存在多个方案，选取使用齿轮数最少的，若还存在多个方案，选取字典序最小的。3<=N<=6，不超过 1000 组数据。	本题显然是搜索题，但由于数据组数较多，需要加入很多优化才能在时限内出解，具体有： （1）容易发现转轴之间形成了一棵树，根结点为主动轴。因此，先枚举转轴形成的树的形态，再枚举每个转轴上套哪些齿轮以及齿轮之间的咬合情况，速度最快（因为可以在枚举过程中及时大力剪枝）； （2）根据题意，这棵树上最多只有一个度为 2 的结点，其它结点的度均为 1。此外，叶结点必然作为时针或分针轴，因此发现某个叶结点的转速既不满足时针轴要求也不满足分针轴要求，就可以剪枝了； （3）在枚举每个转轴上套哪些齿轮时，若这个结点的度为 x，则当该结点是根结点（主动轴）时，最多只能套 x 个齿轮，否则最多只能套(x+1)个齿轮； （4）设使用转轴的个数为 m，如果枚举了一些转轴上套齿轮的情况之后，发现剩下的齿轮个数已经小于剩下的转轴个数，或者已经枚举了(m-1)个转轴后，时针轴和或分针轴都还没有出现，则剪枝； （5）枚举每个转轴上套的齿轮集合的时候，最好压位，这样可以将套上 1 个、2 个和 3 个的方案分开； （6）在发现了合法方案，求出其待求字典序的字符串时，最好将其逆序，即*放在最后面而不是最前面，这样在字符串连接的过程中可以省去一些时间。	时间复杂度： 由搜索量而定 空间复杂度： $O(N\cdot 2^N)$ （由于压位）
2006 G	Pilgrimage	有一个旅行过程中的账本，其中有 4 种可能的事件： <1>PAY x，表示目前的所有人总共付了 x 元；	注意到 COLLECT 没有意义，而 PAY 可以合并（因为只有在 IN 和 OUT 时才进行平均分配，所以两次 IN 或 OUT 之间的所有 PAY 都应该合并成一个），显然，当且仅当满足在每次 IN 或 OUT 发生时，PAY 的总钱数可以被目前的人数整除，并且任意时刻的人数都	时间复杂度： $O(N\sqrt{N})$ 空间复杂度：

		<2>COLLECT x，表示目前的每个人都获得了 x 元； <3>IN x，表示来了 x 个人； <4>OUT x，表示走了 x 个人。 在每次人数变化（增加或减少）时，都要对目前的总资产进行平均分配（来人时，来的每个人将获得和原有的人的平均资产相等的钱，走人时，他们会将钱带走）。已知每次平均分配时都能保证所有人的钱都是整数，且任意时刻至少有一个人参与旅行。现给定这个账本中的一页，求出，在这一页开始时，参与旅行的可能人数（如果解数有限，输出所有解或报告无解，如果解数无限，输出最小的解）。1<=所有的 x<=2000，1<=事件数<=50。多组数据。	为正整数时是可行解。如果有 PAY，则任取一个 PAY，枚举其正因数，再扫描一遍判断是否合法即可，如果没有 PAY，则只要使任意时刻人数都为正的解都可以，扫一遍即可的出最小解的值。 不过，本题有一个难以察觉的陷阱，就是，这是“账本中的一页”！！！也就是说，这一页开头和结尾的 PAY（即第一个 IN 或 OUT 之前的 PAY 和最后一个 IN 或 OUT 之后的 PAY）都不能算，因为这一页的前面一页和后面一页的 PAY 情况未知。	$O(N+x)$
2006 H	Pockets	一张正方形的纸平放在桌面上，被 N 条等距离横线（编号 1～N）和 N 条等距离竖线（编号 1～N）划分为(N+1)*(N+1)的网格。现在将其折 M 次，最后折成一个 1×1 的正方形，每次都是沿一条给定编号的横线或竖线向一个方向（横线为上或下，竖线为左或右）折，保证每次都折成更小的矩形，且任意时刻纸都不会从桌面上拿起，纸的厚度不计。问在最后折成 1×1 的正方形时，从其上、下、左、右四个方向总共能看到多少个开口（可能一个开口能从多个方向看到，此时算多次）。1<=N, M<=64。	本题的难点在于如何维护信息，使得能方便地求出各个方向上开口的个数 。注意到题目中“折叠”的本质是“翻转”——找到折叠沿着的线，然后将其上/下/左/右部分以这条线为轴翻转。所以，很容易想到维护”平面纸“，即将目前的纸的所有连接处裁开后的每张纸（必然是矩形，所以可以用 Xmin、Ymin、Xmax、Ymax 四个坐标来描述）。然后，为了能得到原来的纸的样子，还需要维护两个东西，一是”平面纸“之间的连接情况，二是”平面纸“之间的上下叠放情况（即从底部到顶部平面纸的顺序），具体来说，一张平面纸在上、下、左、右边界上都可能与其它平面纸相连接，且最多与一张平面纸连接，故连接情况只要记录每张平面纸在四个边界上各与哪张平面纸（或不与任何平面纸）连接即可。至于上下叠放情况，只需要给这些平面纸确定一个顺序，满足若平面纸 i 被平面纸 j 压在下面，则 i 排在 j 之前即可。维护了这些信息之后，只需要在各个方向上扫描一遍，对于连在一起的平面纸，中间的全部跳过， 就可以得到开口数了。 然后看如何处理所谓的“折叠”（即翻转）操作 。以向上折为例，其它的可以通过将整张纸翻转（向下折）或旋转（向左右折）之后转化为向上折（注意翻转或旋转后，后面的折叠方向也要变化）。向上折，就是将折叠沿着的线下方的部分整体翻转到上方，因此，只需要将下方的所有平面纸逆序加在上方的所有平面纸后面，其它顺序不变，当原来的一张平面纸跨越了折叠沿着的线时，要将其拆分为上下两部分，然后给这两部分之间加入下边界的连接。然后，原来的连接也要有一定改变，具体来说，原来的左或右连接折叠后不变，折叠线上方原来的上下连接也不变，但折叠线下方原来的上下连接由于翻转，要取反（上连接变成下连接，下连接变成上连接）。注意，在每次折叠之后，要对所有的坐标进行重置（即平	时间复杂度： $O(N^2M^2)$ 空间复杂度： $O(NM)$

			移，将最小 X、Y 坐标均改为(0, 0)，消除负坐标），不然可能爆 int。	
2007 C	Grand Prix	在一个倾角为 θ 的斜坡上，建立 xOy，满足 x 轴方向为沿斜坡向上，y 轴方向为沿斜坡的边界地面。给出在这个斜坡平面上的若干个点，编号 0~N（点 0 始终是原点），求出是否可以通过将这些点绕原点旋转一个角度（顺时针、逆时针均可，也可不旋转），使得旋转后这些点的高度随标号不减？旋转的最小角度（包括顺时针和逆时针）是多少？1<=N<=10000，0<=θ<=45°。多组数据。	简单的立体几何题。注意到如果将斜坡平面上的点的坐标对应到整个三维空间中，则点(x, y)对应的坐标是(xcosθ, y, xsinθ)。然后，枚举每两个编号相邻的点 i 和(i+1)，求出当 i 的高度（z 坐标）不大于(i+1)时，旋转角（这里先只考虑逆时针旋转）的范围区间，具体的推导是很容易的，这里省略，容易证明，这个区间的长度一定是 180°（π）。然后，求出这些区间的交集即可。这些区间的交集一定是一段区间，所以可以直接求交集，也可以求这些区间取反后的并集（对于环形区间来说，并集比交集好求，因为可以拆掉跨越了 2π 的区间）。	时间复杂度： $O(N)$ 或 $O(NlogN)$ 空间复杂度： $O(N)$
2007 E	Collecting Luggage	给出平面上的一个简单多边形，现有一个动点，从该多边形的一个指定的顶点出发，沿着多边形的边逆时针匀速运动，有一个人从多边形外的一个指定点出发，去追赶这个动点（人可以以任意速度运动，但不能超过其最大速度，其最大速度大于动点速度），人不能穿过多边形（但可以沿着多边形的边界运动），求出至少经过多长时间人可以追上动点（即与动点位于同一位置）？ 3<=多边形边数<=100，所有点的坐标绝对值不超过 10000，不超过 10 组数据。	由于人的最大速度大于动点速度，所以如果人能在时刻 t 与动点位于同一位置，则对于任意大于 t 的时刻 t’，人也可以在时刻 t’与动点位于同一位置（只需要在时刻 t 追上动点后，和动点一起运动即可）。所以，本题满足二分性质。二分时刻 t，找出动点在 t 时刻到达的位置，接下来的问题就是，求出人是否能在 t 时刻内走到这个位置。 将多边形的每个顶点以及人的起点、动点在时刻 t 的位置这(N+2)个点作为图中的点，如果点 i 和点 j 之间的线段不与多边形相交（在边界上不算相交），则在图中连边(i, j)，建立无向图，然后在这个图中求 S-T 最短路（S 和 T 是什么就不用说了）即可。问题是如何求一条线段是否与一个简单多边形相交？传统方法需要判断各种特殊情况，比较麻烦，好的方法是用三角剖分解决——将多边形剖分成 N 个带符号三角形，然后求出该线段在每个三角形内的部分的长度，加上符号后求和即可，而线段和三角形的相交部分是很好求的（交点只可能是线段两端点、三角形的顶点和线段与三角形边的内交点这 8 个点），注意，如果线段与三角形中属于原多边形边界的那条边重合，则由于边界不算内部，所以当该三角形符号为正时，应该不算，而该三角形符号为负时，应该算-1。当然，和前面的 1998 F 一样，一开始仍然要将所有的点都平移到第一象限。 此外，建图时可以进行一个优化：由于每次二分只有汇点 T 的位置不同，所以可以预处理求出其它(N+1)个点之间的各边，然后每次二分时花在建图上的总时间就由 $O(N^3)$ 变为 $O(N^2)$ 。	时间复杂度： $O(N^3)$ $+O(N^2 logMAXV)$ 空间复杂度： $O(N^2)$
2007 F	Marble Game	在一个 N×N 的正方形网格木板上有 M 个球和 M 个洞，球和洞都以 1～M 编号，此外还有一些隔板位于相邻的格子之间。现在要通过若干次移动操作将所有的球都移到其对应编号的洞里。移动规则为： （1）每次操作可以让所有未掉进洞里的球都朝着一个方向（上或下或左或右）滚	由于 N<=4，可知 M<=8，这样一来，合法的状态总数是不多的（容易验证当 N=4、M=7 时，合法状态总数为 10! / 3!=604800，为最大值），因此，可以将所有合法状态都存下来，用 BFS 解决，问题是 BFS 过程中的判重。一种做法是使用 set 或 hash_set，但这样的速度太慢（本题数据弱，用 set 已可过）。更快的做法是标号法，先预处理出所有可能的合法状态（枚举每个球在哪里即可），然后将它们按字典序排序，得到标号，这样接下来的 BFS 过程中就可以直接	时间复杂度： $O\left(\frac{(N^2-M+1)!}{(N^2-2M+1)!}\cdot N^3\right)$ 空间复杂度：

		动，一个球会一直滚动，直到它撞到边界、隔板或其它球，或者掉进洞里。球不能越过边界、隔板、其它球和未被填充的洞； （2）当球掉进洞里后，会填充上这个洞，之后这个格子将被视为空格，其它球可以越过或在上面停留。掉进洞里的球永远不能离开这个洞； （3）任何时候，一个格子里或一个洞里最多只能有一个球。 求出至少多少次操作可以完成（让所有的球都掉进与其编号相同的洞里），或者报告永远不能完成。2<=N<=4，保证一开始没有球在洞里。多组数据。	用标号进行判重，没必要再开 set，在将状态转化为标号的时候可以用二分查找实现。 不过，本题的真正问题在于它的实现很困难，因为游戏规则中的特殊情况较多，很容易漏掉特殊情况而出错，具体来说有以下两种： （1）如果用球所在的位置（包括洞里的球）来表示状态，就会出错，因为这样会导致一个格子里可能有两个球（一个洞里一个洞外）。所以，正确的方法是，将掉到洞里的球用一个特殊的坐标（比如(-1, -1)）来表示，然后状态转移时特判； （2）在状态转移的时候，要将原状态用一个数组存下来，然后后续的操作都在这个数组上进行，不能再引用原状态的值（可能原来的洞被填充）。	$O\left(\frac{(N^2-M+1)!}{(N^2-2M+1)!}\cdot(N$
2007 G	Network	将 N 条信息拆成 M 个片段，打乱顺序后通过一个缓冲区。信息片段可以在缓冲区内暂存，以后再从缓冲区输出，也可以不进入缓冲区直接输出。现在要求输出的片段中，各条信息都必须按照原来的顺序连续输出（即对于每一条信息，其各个片段都必须按照原来的顺序连续输出，但各条信息之间的输出顺序不限），求出缓冲区的大小至少是多少。1<=N<=5， 1<=M<=1000，不超过 10 组数据。	本题的题意比较容易理解错，注意它只要求“各条信息按照原来的顺序连续输出”，也就是对于来自同一条信息的两个片段 i 和 j，如果在原信息中 i 在 j 前面，则 i 要在 j 前面输出，且 i 和 j 之间不能有来自其它信息的片段，然而，对于信息与信息之间输出的顺序则不限。这样，一开始实际上需要一个枚举——枚举各条信息输出的顺序。然后模拟一下这个输出过程即可，注意需要一个 set 来模拟缓冲区，每次当有新的片段直接输出时，要将 set 中的所有应该紧接着输出的片段都输出。枚举各种方案，求出需要的缓冲区大小的最小值即可。	时间复杂度： $O(N!\,MlogM)$ （logM 主要是由于 set 的内部操作） 空间复杂度： $O(N+M)$
2007 H	Raising the Roof	给出三维空间中的 N 个点以及以这些点为顶点的 M 个三角形，任意两个三角形都不相交或重叠（但可能在边界上相邻）。求出从 z=+∞处向下看能看到的部分的实际总面积（不是投影面积）。3<=N<=300， 1<=M<=1000。多组数据。	本题无法用剖分法解决，由于形成的平面图可能有洞，也不推荐用建平面图求区域的办法来求，因此只能用扫描线了。先求出每个三角形在 xOy 上的投影（一定也是三角形，如果不是三角形，则这个三角形对答案无贡献，可以舍弃），然后在每个顶点与两边交点的 X 坐标处都设置一条扫描线，这样任意一个三角形（指投影的，下同）在相邻两条线之间的部分都是梯形（可能退化成三角形），由于梯形的面积为两底边长之和的一半乘以高，所以只要知道每个三角形被每条扫描线所截的能看见部分的长度，乘以该线两条相邻的线之间的距离的一半，即为其贡献的效果值，累加即为这个三角形能看到的部分的投影总面积，除以这个三角形所在平面与水平面夹角的余弦值，即为实际面积。 需要注意的是，如果用扫描线本题中，不允许出现某条三角形的边与 Y 轴平行的情况，因此一开始要先将所有的三角形旋转一个角度，使得任意一条边都不与 Y 轴平行。至于这个旋转的角度，可以先将所有边的极角列出来，排序，找到相差最大的两个相邻极角，取其平均值，再用 $\frac{\pi}{2}$ 减去它，作为旋转的角度。	时间复杂度： $O(N^2\,MlogM)$ 空间复杂度： $O(N^2)$

2008 G	Net Loss	有一个 N 次函数 y=f(x)（-1<=x<=1），用一个估计函数 y=g(x)去逼近它。已知 y=g(x)是由两条线段组成的： (1)$y=a_1x+a_0$，-1<=x<=c (2)$y=b_1x+b_0$，c<=x<=1 显然这两条线段在 x=c 处相交。 定义 f(x)和 g(x)的距离为： $\int_{-1}^{1}(f(x)-g(x))^2dx$ 现给出 N、c、f(x)各次项系数，求出使得 f(x)和 g(x)距离最小的函数 g(x)，输出其四个系数的值，要求精确到 0.001。 1<=N<=10，-1<c<1，f(x)各次项系数都在[-1, 1]范围内。不超过 5 组测试数据。	本题需要确定估计函数 g(x)在三个点处的函数值，即 g(-1)、g(c)、g(1)的值，才能完全确定 g(x)的解析式，进而完全确定它与原函数 f(x)的距离。关于这三个值，由于是 double，直接枚举肯定不行，好的办法是使用随机化算法，比如模拟退火。 模拟退火的时候，一般是从原解出发，扩展新的解（新解与原解之间的距离应该与目前的温度 T 成正比，即随着 T 的减小而减小），如果新解比目前的最优解更优，则更新目前的最优解，再看它是否能替换原解（如果比原解优，则能替换，否则，设它的值与原解的值之差为 Δv，则以 $e^{\frac{\Delta v}{T}}$（越大越优）或 $e^{\frac{-\Delta v}{T}}$（越小越优）的概率接受新解）。问题是在本题当中，$e^{\frac{-\Delta v}{T}}$ 这个概率太大，容易丢失较优解，所以要减小这个概率，一种常见的方法是将指数乘以一个较大的常数，即 $e^{\frac{-\Delta v W_0}{T}}$，这里的 W_0 可以取 100000～300000。此外，还可以通过调整初始温度、退火速度和退火终止温度，以及尝试取不同的初始解多次进行模拟退火的办法，来得到更优的解。	时间复杂度： $O(SN)$ （其中 S 为迭代次数） 空间复杂度： $O(N)$
2008 H	Painter	给出平面上的 N 个三角形，求出（1）是否有两个三角形相交（有公共点就算相交）；（2）若第（1）问的答案为“否”，输出被包含的层数最多的三角形的层数。 1<=N<=10^5，坐标绝对值不超过 10^{-6}。	先看第一问。如果某两个三角形有公共端点，则第一问答案显然为“是”，否则，将每个三角形的三边都“去掉端点”（准确来说是在两端点处收缩一个很小的距离），然后第一问就转化为求这些线段是否有公共点，这个可以用扫描线解决——当某条线段插入和删除时，都判断它和相邻的线段是否相交。 再来看第二问。仍然用扫描线解决，将三角形分为上凸壳和下凸壳，并且上凸壳的权值为-1，下凸壳为 1。每当一个三角形插入（扫到左端点，上下凸壳同时插入）时，就求出其前面（Y 坐标比它小的）所有凸壳的权值之和，即为其层数。然而，set 并不支持“查询前缀和”这样的操作，难道需要手写平衡树？一种好的办法是，直接找到其上一个凸壳，然后它的层数等于这个凸壳的层数加上这个凸壳的权值，而“找上一个”的操作，set 是可以支持的（lower_bound 或 upper_bound）。	时间复杂度： $O(NlogN)$ 空间复杂度： $O(N)$
2008 I	Password Suspects	给出 M 个只含小写字母的字符串，求同时包含这些字符串（即这些字符串都是它的子串）的长度为 N 的字符串有多少个，如果不超过 42 个，则按字典序依次输出它们。1<=N<=25，0<=M<=10，1<=所有子串长度 len<=10，答案不超过 10^{15}。	处理多子串的问题一般用 AC 自动机。本题在建立 AC 自动机后，可以在上面进行递推，从而得到结果。具体来说，设 S[i]为 AC 自动机的结点 i 的包含的子串后缀集合（即哪些输入中给定的子串是从根到 i 的路径形成的字符串的后缀，这个可以通过沿 Fail 指针上溯得到），F[i][j][s]为长度为 i 的字符串，匹配到 AC 自动机的结点 j，已经出现的子串集合为 s 的个数，则可以简单递推得到结果。问题是输出具体的字符串怎么办。注意本题不能在递推的时候预存结果，即使是预存各个前趋状态也不行，都会 MLE！！正确的实现方法是，如果最后合法字符串不超过 42 个，就往回重新递推一遍，枚举可能的前趋状态（用 DFS 实现），即“时间换空间”的做法。	时间复杂度： $O(NMlen\,2^M\,S)$ （其中 S 为字符集大小，本题中为 26） 空间复杂度： $O(NMlen\,2^M)$
2008 J	The Sky is the Limit	给出平面内的 N 个等腰三角形（底边均位于 X 轴上，顶点均位于 X 轴上方），求出	只需要求出每条腰未被别的腰挡住的部分的总长度即可，即求平面内两条线段的遮挡情况。显然只有在 X 坐标相交部分可能发生遮	时间复杂度： $O(N^2)$

		从 y=+∞处向下看能看到的总长度。1<=N<=100。	挡，对这一部分，如果不相交，上面的挡住下面的，如果相交，则在交点的左右两部分，各有一条线段被挡住。	空间复杂度：$O(N)$
2008 K	Steam Roller	给出一个 N×M 的网格，每条边都有一个通过时间（当通过时间为 0 时表示这条边不可通过）。现在要从一个指定的起点 S 沿着网格边走到一个指定的终点 T（S 和 T 为网格的格点），要求，第一条边和最后一条边需要加倍，中间如果出现转向，则转向前的那条边和转向后的那条边都需要加倍。当一条边被加倍时，其实际消耗时间等于通过时间的两倍，否则（未被加倍时）实际消耗时间等于通过时间。找到从 S 到 T 最快的路径，输出其消耗的总时间，或者报告路径不存在。1<=N, M<=100，0<=所有通过时间<=10000，不超过 21 组数据。	本题的状态很容易想到：格点位置+上一条边的方向，但要注意，上一条边可能已经加倍，也可能没有加倍，而这显然对本状态的转移代价（边权）有影响，所以这个因素也应该加入状态中，进一步，对于初始状态，其”上一条边“不存在，此时”上一条边的方向“应该取和四个方向都不同的一个值（比如取 4，四个方向依次取 0～3）。所以，完整的状态用一个四元组(x, y, k, FF)来表示，其中(x, y)为位置，k 为上一条边的方向（0～4 分别表示上、下、左、右、不存在），FF 表示上一条边是否加倍（当 k=4 时，FF 无意义，不妨取 0）。 接下来就是以这些状态为顶点，状态转移为边求 S-T 最短路（这里 S 为初始状态，T 不唯一，所有能到达终点的状态都是终止状态）。不过要注意对于”第一条边“和”最后一条边“的判断，由于起点和终点可能多次经过，因此不能认为从起点出发的边就是第一条边、到达终点的边就是最后一条边，应该这样判断：其最短路长度 dist=0 的状态引出的边为第一条边，而对最后一条边的判断，则要在最短路求完后处理所有的终止状态时，再算上（未加倍的加倍）。	时间复杂度：$O(NM)$ 空间复杂度：$O(NM)$
2009 B	My Bad	给出一个由 AND 门、OR 门、XOR 门、NOT 门组成的无环逻辑电路，其中的最多一个门可能出了故障，故障的类型有总是输出和正确的相反、总是输出 0、总是输出 1。现在给出若干组输入和对应的输出，求电路中是否有故障，若有，找出哪个门出了故障以及故障类型，也有可能不能判断出来。1<=输入数量 N<=8，1<=门数量 G、输出数量 U<=19。多组数据。	由于逻辑电路无环，先对其进行拓扑排序（门 i 以门 j 的输出作为输入，则图中有边<j, i>），然后，先看无故障的时候输出是否合法，若合法则无故障，再枚举出故障的门以及故障类型，看是否合法。注意，结果为“不能判断”有两种可能，一是无解，二是多解。	时间复杂度：$O(G(G+U)B)$ 空间复杂度：$O((N+U)B)$ B 为验证使用的输入和输出组数。
2009 C	The Return of Carl	有一个正八面体，边长为 10，现给出起点和终点，求出从起点出发沿着正八面体的表面走到终点的最短路径长度。点用两个角度给出，分别表示以正八面体中心 O 为原点建立 Oxyz 时，它在 xOy 上的投影的极角，以及从 O 到它的向量与 Z 轴的夹角。不超过 10^5 组数据。	本题的最好做法是将整个正八面体的所有可能的展开图全部表示在平面上，然后将三维点对应到展开图中的二维点，求最短距离即可。注意，若起终点在正八面体的同一个面内或者有公共边的面内，则只有一种走法，若在没有公共边但有公共顶点的面内，则有两种可能的走法，若在没有公共顶点的两个面内（即对面），有六种可能的走法，需要逐一判断并取最优解。	时间复杂度：$O(1)$ 空间复杂度：$O(1)$
2009 D	Conduit Packing	给出 4 个指定半径的圆，求出它们的最小包围圆（最小的满足这 4 个圆能两两不重叠地放进其内的圆）的半径大小。0<半径<=20000，不超过 100 组数据。	仔细分析之后可以发现，必然存在一种放置方案，使得（1）放入包围圆（以下称为“大圆”）的第一个圆必然与大圆内切；（2）从放入的第二个圆开始，每个放入的圆都必然与已有的两个圆相切（如果是大圆则为内切否则为外切）。因此，二分大圆的半径，再枚举放入顺序，再模拟具体放入过程即可。	时间复杂度：$O(1)$ 空间复杂度：$O(1)$

2009 E	Fare and Balanced	给出一个带权 S-T DAG，边权均为正整数，且每条边都在至少一条 S-T 路径上。现要增加一些边的权值，使得所有 S-T 路径的长度相等，且每条 S-T 路径上最多只有一条边的权值被增加。求这样的方案是否存在，若存在，操作后的 S-T 路径长度最小是多少。2<=点数 N<=50000，2<=边数 M<=50000，边权<=1000，不超过 100 组数据。	首先可以证明，如果合法方案存在，则最优方案的 S-T 路径长度就是原图中 S-T 最长路径的长度 maxd（增加边权后，显然不会小于这个长度，如果大于这个长度，说明每条 S-T 路径上都有一条边的权值被增加，则将所有被增加权值的边的增加量减少 1 即可得到更优的合法方案）。 设 dist_s[i]为原图中 S 到 i 的最长路径长度，dist_t[i]为原图中 i 到 T 的最长路径长度，bool fs[i]为原图中 S 到 i 的所有路径长度是否都相等 ，bool ft[i]为原图中 i 到 T 的所有路径长度是否都相等。注意到，若 fs[i]=false，则从 S 到 i 必然有路径被增加权值，若 ft[i]=false，则从 i 到 T 必然有路径被增加权值。因此，如果有某个点的 fs 和 ft 值均为 false，则问题必然无解（因为从 S 经这个点到 T，必然存在至少一条路径上有两条以上的边被增加权值），否则问题必然有解，对所有满足 fs[a]=true 且 fs[b]=false 的边<a, b>，将其权值增加为 maxd-dist_s[a]-dist_t[b]，即为满足题意的图。	时间复杂度：$O(N+M)$ 空间复杂度：$O(N+M)$
2009 G	House of Cards	有 2N 张牌，分为两种花色，每种花色有点数 1～N 的牌各一张。现在两人（分别对应两种花色）用这些牌进行游戏，规则如下： （1）先取出前 8 张牌，从左到右排成/\/\/\/\的形式，构成初始牌阵。第一张牌的花色决定了谁是先手； （2）两人轮流操作，每次操作者取出目前剩下的最前面一张，并进行以下三种决策之一： <1>将这张牌拿在手里（前提是这个人手里原来没有牌）； <2>在他手里的牌（如果有的话）和这张取出的牌中选一张，水平搭在牌阵中的任意一个\/上面，组成一个三角形，另一张牌（如果有的话）拿在手里； <3>将他手里的牌和这张取出的牌一起，以/\（谁左谁右可任意确定）的形式搭在牌阵中的一张水平的牌上面，组成一个三角形（当然，进行该决策的前提是他手里有牌）； （3）每当牌阵中组成了三角形的时候，就计算分值，具体来说，该三角形中哪种花色的多，就给这种花色对应的人加分，加的分值为构成这个三角形的三张牌的点数之和； （4）最后当所有牌取完后结束，结束时，如果某人手上还拿着牌，则当该牌的花色	本题是典型的对抗搜索。对抗搜索与组合游戏（博弈）问题的最大不同是，对于参与游戏的双方来说，规则不同。对抗搜索一般用 DP 解决，即，先“前推后”找出所有可能的状态，再从后往前算出每个状态的值，至于如何找出状态，一般用与插头 DP 类似的方式，即使用一个 map 存储所有状态及其编号，再使用一个数组从编号链接到状态（可以使用 list 或其它链表保存状态之间的关系，避免在 DP 时重算，减少时间消耗）。 回到本题，注意到给对手加分相当于给自己减分，反之亦然，所以，重新定义一个人的分值为”按照题目中的规则，他的分值减去对手的分值“。这样以来，在任何时刻双方的分值都互为相反数，知道了一方的分值，也就知道了另一方的分值。 设 F[i][j]为第 i 阶段（即取了 i 张牌后），出现的状态为 j 时，本阶段的操作者能得到的最高分。则显然有 F[i][j]=max{S[j][k]-F[i+1][k]}，其中 k 为 j 的后继状态，S[j][k]为 j 转移到 k 的代价（效果值，即在转移时形成的三角形导致的分值变化）。若 i=2N（即终止状态），则 F[i][j]为 j 状态本身的分值（由于手上剩下的牌导致的分值变化）。 接下来还有一个问题没解决，状态怎么表示？本题的状态很复杂，可以发现，一个状态需要记录如下信息： （1）目前牌阵中所有”露在外面“的牌的位置关系（用 1、0、-1 分别表示/、-、\）以及这些牌的编号（是原来的第几张牌，注意用编号而不用点数的原因是对于各组数据可以通用，不必每次都重新计算状态，加快速度）； （2）双方手里拿着的牌的编号（同样用编号不用点数）； （3）操作者是谁。	时间复杂度：$O(NS)$ 空间复杂度：$O(NS)$ （其中 S 为每个阶段的状态总数，经实践证明，第 25 阶段的状态总数最多，为 20184）

		与他对应时，他加分，否则减分，加或减的分值为该牌的点数。 给出 N 和 2N 张牌从前到后的顺序，以及其中一个参与游戏的人，求出在双方都采取最优策略的情况下，他最终的分值减去对方最终的分值是多少。5<=N<=13，不超过 5 组数据。		
2009 J	Subway Timing	给出一棵树，每条边有一个时间权值（单位：秒），现在要以分钟为单位对这些边权进行取整，每条边可以选择上取整或下取整到整分钟数，要求，在取整之后，树上的路径的总长度（即总时间）与取整前的误差绝对值的最大值最小能是多少？ 1<=N<=100，1<=边权<=300，边权一定是整数。多组数据。	先二分误差最大值 R（单位：秒），然后通过 DP 求出是否存在使得所有路径误差绝对值都不超过 R 的方案。设 F1[i][j]为对于结点 i，从它到它子树内的任意一个结点的误差（以下称为”向下误差“）最大值为 j（注意，这里的”误差“不取绝对值，也就是 j 也可能是负数），且子树 i 内的所有路径的误差都在[-R, R]范围内时，向下误差最小值最大能是多少，F2[i][j]则表示 i 的向下误差不超过 j，其它条件相同时的对应结果，显然 F2[i][j]=max{F1[i][_j]，_j<=j}。 计算 F1[i][j]时，若 i 是叶结点，则 F1[i][0]=0，其它的 F1[i][j]均为负无穷。若 i 不是叶结点，则枚举 i 的一个子结点 k，并假设 i 的最大的向下误差出现在子树 k 里。显然，i 的其它子树里的最大的向下误差都不能超过 j，同时也不能超过 R-j，所以，对于每个子结点_k，枚举 i 到它的边是上取整还是下取整，即可得到所引用的 F2[_k][_j]中的_j 的值，然后在这两种方案中取向下误差最小值较大的，在子结点之间取这个值的最小值即为 F1[i][j]（注意，如果这个结果的最小值和次小值之和小于-R，则不合法，F1[i][j]为负无穷）。	时间复杂度： $O(NMlogMAXS)$ 空间复杂度： $O(NM)$ 其中 MAXS 为二分范围，M 为最大深度*60，即可能的最大误差。
2009 K	Suffix-Replacement Grammars	给定两个长度均为 N 的字符串 S、T，以及 M 个变换规则，求出至少多少次变换可以将 S 变为 T。每个变换规则由两个等长的字符串 X 和 Y 组成，表示如果待变换的字符串有后缀 X，则可以将这个后缀变成 Y（如果没有后缀 X 则不能使用此规则变换）。1<=N<=20，1<=M<=100。多组数据。	容易发现，本题中可能变换出的字符串总数非常多，远远超过 long long 范围（甚至，变换次数都可能达到 long long 级别），所以基于搜索的算法肯定不可行。 先考虑这样一个问题：给出一个 i，满足 1<=i<=N，求出至少多少次变换才能将 S 的后 i 个字符变换成 T 的后 i 个字符？显然，变换必然是以下步骤：用一些长度为 i 的规则，在每两个长度为 i 的规则之间用一些长度小于 i 的规则。如果将相邻两次使用长度为 i 的规则之间的部分称为“一个阶段”，则在每个阶段的起始和终止时，这个字符串必然都是已有字符串的长度为 i 的后缀（S 和 T，以及规则中出现的字符串都是已有字符串）。所以，将所有长度不小于 i 的已有字符串的长度为 i 的后缀当作点，如果点 a 能通过一次长度为 i 的变换规则变成点 b，则连边<a, b>，权值为 1；如果点 a 和 b 的最左边的字符相同，而 a 的后(i-1)个字符能通过 x 次变换变成 b 的后(i-1)位，则连边<a, b>，权值为 x。显然，这个有向图中从表示 S 的后 i 个字符的点到表示 T 的后 i 个字符的点的最短路径，就是这个子问题的答案。问题是 x 怎么求出？容易发现，当 i=1 时，x=0，而当 i>1 时，x 其实就是长度为(i-1)时的子问题——所以 i 从 1 开始，递推求解子问题即可。为了优化，可以进行去重，将相同的字符串去掉。	时间复杂度： $O(NM^3)$ 空间复杂度： $O(M^2)$

2010 E	Channel	给出一个 N×M 的矩阵，有些格子是障碍格，现在要求找到从左上角到右下角（保证左上角和右下角不是障碍格）的最长的满足以下条件的路径，并输出（有多解时输出任意一条均可）： （1）不能经过障碍格； （2）两个在路径中不相邻的格子不能在矩阵中处于相邻位置； （3）两个在路径中不是只间隔一个的格子不能在矩阵中处于对角相邻位置。 2<=N<=20，2<=M<=9。多组数据。	本题是典型的插头 DP 问题。所谓插头 DP，是指一类需要在状态中维护连通性的状态压缩 DP 问题。插头 DP 的状态表示一般比较复杂，所以也要用 map 来存储。 对于本题，逐格 DP，状态中需要存储的信息有： （1）目前阶段（格子）的前 M 个格子内是否占用，以及它们之间的连通性； （2）对于目前阶段（格子）的前 M 个格子，它们的上方的那个格子是否占用； （3）目前阶段的左上方的格子（如果有的话）是否占用。 然后在 DP 时，判断各种情况即可。注意，由于题目限制了左上角到右下角，且是一条路径，因此某个非左上角也非右下角的占用格只有一个相邻的占用格，不合法，某个连通块中途消失，也不合法。	时间复杂度： $O(NM3^M)$ 空间复杂度： $O(NM3^M)$
2010 F	Contour Mapping	给出一个等边三角形网格图，每个顶点有一个高度，求出这里面所有高度为 h 的倍数的等高线的总长度。1<=h<=1000，三角形不超过 10000 个。	对每个三角形分开处理。对于一个三角形，有三种情况：（1）三个顶点高度都不同；（2）有两个顶点高度相同，且与另一个不同；（3）三个顶点高度都相同。对于情况（1），从高度第二大的那个向其对边上与其高度相同的点连一条线，将整个三角形分成两部分，转化为情况（2）。对于情况（2），直接用类似等比数列求和的办法求等高线的总长度。对于情况（3），只有当某条边的两侧的点（如果存在的话）不都和这个边的高度相同的时候才能计入。	时间复杂度： $O(N)$ 空间复杂度： $O(N)$ 其中 N 为三角形个数。
2010 H	Rain	给出平面上的 N 个指定点及其高度，以及 M 条连接这些点的边，满足它们划分出的每个区域都是三角形（假设这些区域之外的部分都无穷低）。现在需要求出，如果从无穷高处往这里倒水，则在这个区域内可以形成多少个水域，以及每个水域的水面高度。1<=N<=2704。	一开始当然是建平面图求区域。问题是到底如何判断哪些地方能“存住”水，形成水域？容易发现，每个水域中至少包含一个指定点，且满足这个指定点的水面高度大于其本身的高度。进一步，分析水漏出的过程可以发现：一个地方高度为 h0 的水可以漏出，当且仅当图中存在至少一条由这个地方到某个边界上的指定点的路径，满足该路径上的各个点的高度都小于（不能等于）h0。因此，问题转化为先求出边界点（求平面图的区域过程中得出），再计算对于图中的任意一个指定点，它到边界点的最高点最低的路径上的最高点的高度。容易发现这条最高点最低的路径一定是由原图中的边组成的，因此就转化为 SPFA 松弛问题了。	时间复杂度： $O(NlogN)$ 空间复杂度： $O(N)$ （根据平面图的性质，本题中的 M 显然也是 $O(N)$ 级别的）
2011 A	To Add or to Multiply	有一种由 A 和 M 组成的对一个整数的操作序列，A 表示将这个整数加 a，M 表示将这个整数乘 m（a、m 都是指定的正整数）。现在要求出最短的这种操作序列，使得当操作前的整数位于[p, q]范围内时，操作后的整数一定位于[r, s]范围内。1<=a, m, p, q, r, s<=1000000000，p<=q，r<=s，不超过 15 组测试数据。	为了便于描述，以下设 f(x)为对于一个指定的操作序列，当操作前的整数为 x 时，操作后的整数的值。 若 m=1，则 M 操作无用，只需考察仅由 A 操作组成的序列是否有符合要求的即可。 $O(logs)$ 若 m>1，则由于最终结果不能超过 10^9 可知最多有 29 个 M 操作，所以可以枚举 M 操作的个数 n。接下来的问题是确定任意两个 M 操作之间（以及第一个 M 操作之前和第 n 个 M 操作之后）进行的 A 操作的个数。设其分别为 $c_n, c_{n-1}...c_0$。这样： $f(x)=(((x+c_n\cdot a)m+c_{n-1}\cdot a)m+\ldots)m+c_0$ 展开得	时间复杂度： $O(logs)$ 空间复杂度： $O(logs)$

			$f(x)=x\cdot m^n+a\sum_{i=0}^{n}c_i m^i$ 由于对于固定的操作序列，操作后的值随操作前的值递增而递增，所以原题的限制可以转化为 $f(p)\geq r, f(q)\leq s$ 。在枚举了 n 后，可以得到 $\sum_{i=0}^{n}c_i m^i$ 的值的范围，设为[l, r]。容易发现，$(c_n, c_n-1, \ldots, c_0)$ 就是这个值的 m 进制表示！准确来说，要在[l, r]中找到一个整数，使得它的 m 进制表示式的(n+1)位数（超过(n+1)位则 c_n 可大于等于 m）之和最小。这个数其实很容易找到：l、r 的 m 进制表示的前若干位公共部分，加上 l 和 r 的 m 进制表示的最高非公共部分位的 l 这一位数的结果加 1，后面再加上若干个 0 即可。	
2011 B	Affine Mess	平面上有三个整点，对它们进行一次旋转、一次放大和一次平移操作（旋转操作最先进行，放大和平移操作顺序未知）。旋转时，将 X 轴通过以原点为中心、边长为 20 的正方形上的一个整点，再将所有的点移动到距离它最近的整点处（即四舍五入）；放大时，X 和 Y 坐标的放大倍数都是非零整数（可能不等）；平移时，X 和 Y 坐标的平移量也都是整数（可能不等）。现给出这三个整点操作前的坐标和三次操作后的坐标（均为整数，但操作前哪个点对应操作后哪个点未知），求出：（1）是否存在可行解；（2）若存在，所有的可行解是否完全等价（即对于平面上的任意一个点，不管选取哪个可行解，都会在操作之后得到相同的结果）？输入的所有坐标绝对值不超过 500，且为整数。多组数据。	容易发现，如果先平移再放大是可行的，那么先放大再平移也是可行的（以 X 坐标为例，如果先平移再放大，平移量为 a，放大倍数为 b，则 x 变成 b(x+a)=bx+ab，相当于先放大 b 倍再进行 ab 的平移），因此只需要考虑先放大后平移的情况。 本题只有三个点，因此可以暴力枚举。具体来说，先枚举这三个点操作之前与操作之后的对应关系（哪个对应哪个），再枚举旋转经过的点（即旋转角度）。旋转之后，根据操作完成后三个点的坐标之差和目前三个点的坐标之差的比值，可以得出 X 和 Y 坐标的放大倍数，再根据放大后的坐标差得到平移量，最后验证是否可行即可。注意一种特殊情况：如果目前三个点的 X 坐标都相同或者 Y 坐标都相同，则当操作完成后三个点的 X 坐标或 Y 坐标都相同时，该放大倍数可以取任意值，因此得到的解（如果有的话）一定不等价，而当操作完成后三个点的 X 坐标或 Y 坐标不都相同时，是无解的。	时间复杂度：$O(1)$ 空间复杂度：$O(1)$
2011 D	Chips Challenge	给出一个 N×N 矩阵，要求在其中的每个位置都填上 0 或 1，满足： （1）对于任意的 1<=i<=N，第 i 行的数的和等于第 i 列的数的和； （2）对于任意的 1<=i<=N，第 i 行/列的数的和不超过矩阵中数字总和的 A/B； （3）有些位置已经预先填好了数（0 或 1），不可修改。 求整个矩阵的数字总和最大能是多少。 1<=N<=40。	枚举每行/列和的最大限制 maxv，则问题转化为找到一种填数方案，使得对于任意 1<=i<=N，第 i 行和第 i 列和相等且不超过 maxv，整个矩阵的和最大，然后看这个最大和是否不小于 maxv*(B/A)即可。 在这个新问题中，注意到“第 i 行和第 i 列相等”是一个相等限制，可以联想到流量平衡；而“整个矩阵的和最大”则可以联想到费用因素。所以，建立一个有向图，对于任意 1<=i<=N，图中有 i'和 i"这两个点分别表示第 i 行和第 i 列，连边<i', i">，容量上界 maxv，下界 0，费用 1（传递流量限制）；如果矩阵中(i, j)位置未被预先填数，连边<i", j'>，容量上界 1，下界 0，费用 0；如果矩阵中位置(i, j)已预先填 1，连边<i", j'>，容量上下界均为 1，费用 0。问题转化为求这个图的最大费用可行流。	时间复杂度：$O(N^4)$ 空间复杂度：$O(N^2)$

			先将这个图的所有下界大于 0 的边的容量强制设为它的下界，将它的容量上界减去下界，从而消去下界；然后，要求最大费用可行流，为了避免正环出现，将这个图的所有正费用边强制满流，从而消去正费用边。以上的操作造成了点容量的预先盈余和亏空，所以要设置附加源汇对这些预先盈余和亏空进行处理，用附加源汇增广，求出最大费用最大流，如果能使所有与附加源汇相连的边都满流则原图存在可行流，最大费用也同时得出。 特别注意：本题在枚举 maxv 时不能二分！！	
2011 G	Magic Sticks	给出一个 N 节轴，每节是一条指定长度的线段。可以将这个 N 节轴的某一段进行折叠，折叠成一个多边形，也可以舍弃某些节。给定 N 和这个 N 节轴各节长度，求出，用它折叠成一个或多个多边形后，这些多边形的最大总面积是多少？ 3<=N<=500，1<=各节长度<=1000。	本题的难点在于给定一个多边形各边（按顺序）的长度，求出其最大面积。关键性质：多边形面积达到最大时，一定是圆内接多边形，且给定各边长度后，最多只有一个圆内接多边形。因此，问题转化为二分圆的长度，求是否存在给定的圆内接多边形。 在求是否存在圆内接多边形时，有三种可能，圆心在多边形内、圆心在多边形外、圆心在多边形的一条边上。这个可以通过求圆心到各条边两个端点的夹角和来实现（先设圆的直径为多边形最长的边，如果此时的夹角和大于 2π，则为圆心在多边形内，若小于 2π 则为圆心在多边形外，若等于 2π 则为圆心在多边形的边上），然后二分半径得出解。注意，本题不能采用整体二分，这样更慢！！ 此外，本题不能枚举所有的区间然后求最大面积，因为这样的总时间复杂度会升到 $O(N^3 logMAXS)$，会 TLE。正确做法是只枚举以下的区间[l, r]：<1> r=N-1；<2>[l, r]可以构成多边形（即所有边长度和大于最长边的两倍）但[l, r+1]不能；<3>第(r+1)条边的长度不小于[l, r]中最长边的两倍。 还有就是本题的精度问题也是很烦人的，在二分边界和求夹角的时候，zero 不能取得太小。	时间复杂度： $O(N^2 logMAXS)$ （其中 MAXS 为二分范围） 空间复杂度： $O(N^2)$
2011 H	Mining Your Own Business	给出一个无向图，要求在上面标记一些点，使得在该图中删除任意一点后形成的每个连通块中都有至少一个被标记的点。求出最少要标记多少点，以及最优方案的总数。1<=边数 M<=50000，不超过 20 组测试数据。	容易发现，标记割点显然不合算，而对于原图的每个点双连通分支，只要标记任意一个非割点（如果有非割点的话），就可以满足题目的要求。因此，问题转化为求出图的每个点双连通分支内的非割点个数，用 Tarjan 算法，设置辅助栈保存经过的边，在遇到割点时，设遇到的满足 low[j]>=dfn[i]的边为(i, j)，就将辅助栈中(i, j)及其后面的边都弹出，它们及其关联的点组成一个点双连通分支，两问同时解决。注意一种特殊情况：如果原图就是点双连通图（即不存在割点），则最少标记点数为 2 而不是 1（一个标记点被删了还有另一个），标记任意两个点均可。	时间复杂度： $O(N)$ 空间复杂度： $O(N)$
2011 I	Mummy Madness	平面上有一个人和 N 个木乃伊，每回合人先走一步（也可以不走），然后所有木乃伊都会走一步，使得它与这个人的 Euclid 距离尽可能小。每一步有 8 种可能(-1, -1)、(-1, 0)、(-1, 1)、(0, -1)、(0, 1)、(1, -1)、(1, 0)、(1, 1)。求出最多多少回合之后人会被木乃伊抓住（人和至少一个木乃伊	由于人不管怎么走，都无法增大与任意一个木乃伊之间的距离，而如果人走某一步之后与某个木乃伊之间的距离减小，则在此之前人与该木乃伊之间的距离必然不断减小（这里的距离为 8 方向距离，即 X 坐标差和 Y 坐标差的较大值）。所以可以得到，如果在 1000001 回合后人仍然可以不被木乃伊抓住，则人可以永远不被木乃伊抓住。 否则，二分回合数 S，判断人在 S 回合之后是否可以不被木乃伊抓	时间复杂度： $O(NlogNlogM)$ 其中 M 为 1000000 空间复杂度： $O(N)$

		位于同一位置则被抓住），或者人可以永远不被抓住。1<=N<=100000，人的初始坐标为(0, 0)，所有木乃伊的初始坐标的绝对值不超过 1000000。	住。可以发现，这个问题等价于以人和每个木乃伊的初始位置为中心画一个边长为 2S 的正方形，人的那个正方形内是否有位置不被任何一个木乃伊的正方形覆盖到。这其实就是矩形面积并问题（将每个木乃伊的正方形与人的正方形的交集拿出来，交集必然是矩形，求这些矩形的面积并是否为 $4S^2$），这是经典问题，可以用线段树+扫描线在 $O(NlogN)$ 时间内解决。注意，由于本题时限较紧，需要使用常数小的 ZKW 线段树。	
2012 A	Asteroid Rangers	给出三维空间中的 N 个点，每个点都在向一个方向做匀速直线运动。任何时刻，两点之间边的权值为两点的 Euclid 距离。求整个运动过程中这 N 个点的最小生成树会改变多少次。保证任何时刻最小生成树均唯一。2<=N<=50，所有坐标的绝对值不超过 150。	设两个点的初始坐标为(x1, y1, z1)与(x2, y2, z2)，速度坐标为(vx1, vy1, vz1)与(vx2, vy2, vz2)，则 t 时刻两点的距离为 $\sqrt{[(x1-x2)+t(vx1-vx2)]^2+[(y1-y2)+t(vy1-vy2)]^2+[(z1-z2)+t(vz1-vz2)]^2}$ 其中根号下的部分是关于 t 的二次函数（也有可能是一次函数），这样它们之间的距离呈单峰（抛物线）。枚举任意两条边，可以计算出这两条边对应的抛物线（或直线）的交点（最多两个），由于只有在交点处会发生两条边的边权大小改变的情况，因此只有在交点处，MST 可能改变。故先预处理求出所有交点与初始的 MST，然后从前到后枚举交点，在某个交点处，设边 e1 的权值开始小于边 e2，则尝试在原来的 MST 中删去 e2，加入 e1，看是否还能得到树，若还能得到，则 MST 改变一次。特别需要注意的是，如果多个交点事件同时发生，则只能计算一次！！该算法的时间复杂度，看上去是 $O(N^5)$（交点个数为 $O(N^4)$，每个交点事件需要 $O(N)$ 的时间处理），实际上只有 $O(N^2)$ 个交点事件满足在该事件发生时，e1 在 MST 中而 e2 不在，所以总的时间复杂度是 $O(N^4)$ 的。	时间复杂度： $O(N^4)$ 空间复杂度： $O(N^4)$
2012 C	Bus Tour	给出一个 N 个点（点从 0 到(N-1)编号）的无向图，求出一个用时最短的访问序列，要求：（1）从 0 号结点开始，访问 1～(N-2)号结点各一次后到达(N-1)号结点，然后再访问 1～(N-2)号结点各一次后返回 0 号结点；（2）在第一次访问中，1~(N-2)号结点中前(N/2)（下取整）个访问的结点在第二次访问中也必须是前(N/2)（下取整）个访问；（3）访问结点 i 后紧接着访问结点 j 的时间为图中 i 到 j 的最短路径长度。求最短访问时间。 1<=N<=20，图中至少有两条边。	搜索。注意限制条件（2）给了我们一个显然的思路：枚举所有(N/2)（下取整）与(N/2)（上取整）个点的访问方案，得出最短访问时间，再合并成一条路径。这里需要采用分段搜索的思想：先得到所有 i 个点的访问方案（可能从 0 或(N-1)开始，不包括 0 或(N-1)）的最短时间（按照其最后一个访问的结点分类，状态压缩），再由此得出所有(i+1)个点的访问方案的最短时间。	时间复杂度： $O(NC_N^{N/2}+N2^N)$ 空间复杂度 $O(NC_N^{N/2}+N2^N)$
2012 E	Infiltration	给出一个 N 个点的有向图，每两点间都有且只有一条边。现在要求选出一些点，使得图中的每个点要么被选出，要么至少一个被选出的点向它连有边。求至少要选多少个点，并输出一种方案。1<=N<=75。	DLX（Dancing Link X)算法解决重复覆盖问题。 本题可以抽象成如下问题：一个 01 矩阵，要求选出一些行，使得对于每一列，都有至少一个选出的行的该列元素是 1。这个问题称为重复覆盖问题，可以用 DLX 算法解决。对这个矩阵建立一个二维 Dancing Link，每个 1 指向它的正上、正下、正左、正右方向离	时间复杂度： 由搜索量而定 空间复杂度： $O(N^2)$

			它最近的一个 1（加入行头和列头，如果某个 1 不存在则指向行头或列头）。在这个图中进行搜索，每次找到那个 1 最少的列，在覆盖了这一列的行中选一个，并把这一行覆盖的所有列删除。加入启发式优化后，可以很快得到解。	
2012 G	Minimum Cost Flow	给出一个 N 个点、M 条边的无向图，每个点有一个坐标(x, y, z)（其中 z 表示它的高度），某些点上可能有一个或多个洞。现在要向这个无向图里注水，使得水可以从一个指定的原点 S 流到一个指定的汇点 T，且满足以下条件： （1）水压（决定水能到达的高度）可以任意确定，确定了水压之后，所有高度不高于水能到达的高度的点，都可能有水流入； （2）水不能经过一个有洞的点，因此需要用塞子将某些点上的洞堵上，一个塞子的费用是 0.5； （3）可以在两个点间连接新边，前提是这两个点上都至少有一个开着的洞。加入这条新边的费用为两点距离，连接新边后，新边会将两个洞都堵上； 求出满足条件且花费最少的方案。边与边之间即使相交也不会关联 。2<=N<=400，0<=M<=50000，0<=一个点上洞的个数<=400，所有坐标绝对值不超过 10^4。多组数据。	本题一看就知道是图论中的最短路模型，问题是具体的模型很难建立。如果对图的每个连通块为缩点，则对于”一个（原来的）点上若只有一个开孔，就不能在这个点上加两条新边”这种情况无法处理；而如果保留原图的所有点不变，则对于这种情况也不好处理。正解是，将所有至少有两个开孔的点都拆成两个点，与之关联的边当然也要复制，同时在这拆成的两个点中加一条权值为 0 的点，原图中的所有边（包括复制后的边）权值也都为 0。然后，限制每个点最多只能与一条新加边相连。这样在求最短路的时候就需要记录两个最短路值 D1[i]和 D2[i]，分别表示 i 的上一条边是新边和不是新边的情况。由于使用两个最短路值同时更新，所以不能再用 Dijkstra 等固定性算法求最短路，只能使用松弛性算法（Bellman-Ford 和 SPFA，推荐 SPFA）。 这样做时间复杂度是三方的，而且是拆点之后，因此不进行常数优化很容易 TLE。常数优化主要也体现在“拆点”上——容易发现，同一个点拆成的两个点，其所属的连通块一定相同，而且其 D1 和 D2 两个最短路值也一定相同（因为这两个点其实是等价的），这样，在求连通块以及求最短路的之后，只需要算出一个点的值，就可以同时得到另一个的值，从而减少一半的常数，就可以 AC 了。	时间复杂度： $O(N^3+M)$ 空间复杂度： $O(N^2+M)$
2012 I	A Safe Bet	一个 R×C 的网格中有 N 个镜子，镜子有/和\两种朝向，一个格子里最多只有一个镜子。光线射入镜子后会被镜面反射，如果镜子的构造满足从网格左上方向右射入的光线最终能从网格右下方向右射出，则这种构造是”可通过“的。现在给出 R、C、N 以及各个镜子的位置和朝向，求出：（1）这种构造是不是可通过的；（2）如果（1）的答案为“否 ”，求出网格中有多少个位置满足，在此加一个镜子（朝向任意）能够使构造变成可通过的。1<=R, C<=10^6。45%的测试点 1<=N<=12000，不超过 32 组数据，剩下的测试点 1<=N<=4*10^5，不超过 10 组数据。	本题需要发现一个重要的性质，即“两条从外面射入且初始来源或方向不同的光线，不会在里面汇聚（即同向到达同一个格子）”，同时，这个性质也可以推导出，从外面射入的光线不会在里面出现循环而永远出不来，所以不用再考虑循环的问题。有了这个性质，就很容易设计出算法——找到从左上角向右射入的光线和右下角向左射入的光线，在里面经过的每条水平线段和竖直线段（有镜子的端点不计），再转化为求一些水平线段和一些竖直线段的交点个数问题。 求水平线段和竖直线段的交点个数，显然需要使用数据结构，但使用什么数据结构呢？其实借助扫描线，用树状数组就可以解决。设置一个数组 A，将竖直线段取出插入事件（上端点）和删除事件（下端点），设该竖直线段的列号为 i，则插入时 A[i]加 1，删除时 A[i]减 1，扫描到水平线段时，设其跨越的列号为[l, r]，求 A[l..r]的和即可——这显然可以用树状数组做。 此外，本题需要特判 N=0，即没有镜子的特殊情况。	时间复杂度： $O(N(logN+logC))$ 空间复杂度： $O(N+C)$
2013 A	Self-Assembly	给出 N 种正方形，每种正方形四条边上各	首先可以证明，能拼成一个无限大的图形，当且仅当能只用“将一个	（算法二）

		有一个标识，标识有两种：一个大写字母(A~Z)加一个'+'或'-'，或者”00”。两个正方形可以拼在一起，当且仅当它们重合的那条边上的标志为字母相同，符号相反，”00”不能与任何边拼在一起。正方形可以翻转或旋转，每种正方形可以使用无限次，求能否用这些正方形拼成一个无限大的图形？1<=N<=40000。多组数据。	新的正方形接在上一个正方形的上方或右方”的操作拼无限次。 算法一：预处理出由原来的 N 种正方形翻转或旋转得到的 8N 种正方形，如果 j 可以接在 i 的上方或右方，连边<i, j>，形成一个有向图，然后判断有没有环即可。算法一的时间复杂度可能达到 $O(N^2)$，无法通过本题； 算法二：可以使用“插头”的思想。一开始将所有的标志（共 53 种）拆成 4 个点，分别表示该标志出现在某个正方形的上下左右边上。每个正方形的下边标志向上边标志、左边标志向右边标志、下边标志向右边标志、左边标志向上边标志各连一条有向边，如果 j 能接在 i 的上方，则 i 的上边标志向 j 的下边标志连一条有向边，如果 j 能接在 i 的右方，则 i 的右边标志向 j 的左边标志连一条有向边，判断这个图有没有环即可。算法二的时间复杂度为 O(N)。	时间复杂度 $O(N)$ 空间复杂度 $O(N)$
2013 C	Surely You Congest	一个带边权的无向图，边权均为正，某些点上有一些车，现在要使尽可能多的车开到结点 1，要求同时开出，必须走最短路，且任何时候不能有两辆车同时出现在同一条边的同一位置，问最多能让几辆车开到点 1。1<=点数 N<=25000，0<=边数 M<=50000，0<=车的总数<=1000。	简单分析可以得出，如果两辆车所在的点到 1 的最短距离相等，则它们不能经过同一条边，否则必然会撞上（因为都走最短路，如果到达了同一条边，必然是同时到达），而当两辆车所在的点到 1 的最短距离不等时，它们即使经过了同一条边也不可能撞上。这样，可以把所有点按照到 1 的最短距离分类，对每一类分开处理。 因此，先求出点 1 到所有点的最短距离，设 1 到 i 的最短距离为 dist[i]，这一步可以用 Dijkstra 堆优化或 SPFA 在 O((N+M)logN)或 O(M)时间内完成。然后构造原图的最短路图（SPT），即由原图所有点与所有满足 dist[i]+w<i, j>=dist[j]的边组成的图（是有向图）。由于在 SPT 中怎么走都是最短路，因此，对于上面所说的每一类点，只要它们上面的所有车两两之间不走同一条边即可。构造汇点 T，这一类的所有点向 T 连一条边，容量为该点上的车的数量。求从 1 到 T 的最大流，就是最多能开出的车数，累加各类的结果即为最终结果。	时间复杂度： $O(M+cM)$ （SPFA+Dinic）或 $O((M+N)logN+cM)$ （Dijkstra 堆优化+Dinic） 空间复杂度： $O(N+M)$
2013 D	Factors	记一个正整数 k 的质因数分解式的不同排列方式的数目为 f(k)（如 f(10)=2，因为 10=2*5=5*2），给出 n，求最小的满足 f(k)=n 的正整数 k。1<=n, k<2^{63}。不超过 1000 组数据。	设正整数 k 的质因数分解式中，各个质因数分别有 $a_1,a_2,...,a_m$ 个，则 $$f(k)=\frac{\left(\sum_{i=1}^{m} a_i\right)!}{\prod_{i=1}^{m} a_i!}$$ 故只需要枚举 m 与 $a_1,a_2,...,a_m$ 的和，分子就可以确定，又由于 f(k)已知，分母也可以确定了。接下来只需要枚举各个 a_i 即可，在枚举的时候必须保证倍数限制（即分母始终是这些 a_i!的积的倍数）。为了方便处理大数字，可以预处理出 1 到 63 范围内的所有质数，用它们的指数的乘积来表示阶乘，这时可以在枚举过程中加入一个很强的剪枝：如果目前剩余的和为 S，剩余的乘积值为 M，则当 S!的某个质因数的指数小于 M 时，就肯定无解，可以剪枝。加入这个剪枝之后就可以很快地出解了。	时间复杂度： 由搜索量而定 空间复杂度： $O(Slogk)$，其中 S=18，为 1～63 范围内的质数个数
2013 E	Harvard	给出一个程序，包括变量调用操作和循环	先枚举存入 0 号内存块的变量（容易证明最优方案必然将 0 号块装	时间复杂度：

		操作（无空循环，可能有嵌套循环），有 b 个内存块，编号 0~(b-1)，每个内存块内最多可以存 s 个变量。在调用变量时，0 号内存块内的变量可以直接调用，其它内存块内的变量则需要通过 BSR 来调用，BSR 是一个整数，表示目前可以读取的内存块的的编号。显然，当目前所需要调用的变量既不在 0 号内存块也不在 BSR 指定的内存块中时，需要在调用之前将 BSR 改为该变量所在的内存块编号。BSR 的值一开始未定义，必须要对其至少改值一次后才能读取。现在要求给变量分配内存，使得执行整个程序进行的操作（包括调用变量和改变 BSR 的值）总次数最少。1<=b, s<=13，1<=变量个数<=min(b*s, 13)，程序的长度不超过 1000，单个循环的循环次数不超过 10^6，执行整个程序调用变量的次数不超过 10^{12}。	满）。然后扫描整个程序，将其中所有 0 号内存块的变量删去，然后求出剩下的变量中的所有 S[i][j]值，表示调用 i 后面紧接着是调用 j 这样的次数。具体实现时，从内到外扫描所有的循环，计算每个循环内的所有 S[i][j]值和每个循环第一个、最后一个调用的非 0 号块变量。得到 S 数组之后，再枚举其它的变量存到哪里（注意非 0 号内存块不是必须要装满，有可能不装满更优），用 A 数组的值得到需要改变 BSR 值的次数，中间加入适当的剪枝即可。 注意特殊情况：一是 s 大于等于变量个数，只需要将所有的变量全部存入 0 号内存块即可。二是在扫描循环的时候，可能某个循环内一个非 0 号块变量也没有，需要特殊判断一下。此外，在枚举 0 号块变量以前还要进行一次预处理，从内到外计算所有循环的调用变量次数。	$O\left(C_n^s(m+(n-s)!)\right)$ 空间复杂度： $O(n^2m)$ 其中 n 为变量个数，m 为程序长度。
2013 F	Low Power	将 2nk 个正整数分成 n 部分，每部分 2k 个，分成两组，每组 k 个。记每部分的差异值为该部分的两组内的最小值的差的绝对值，现要求给出一种分配方案使得各部分的差异值的最大值尽可能小，输出这个最小的最大差异值。1<=2nk<=10^6，所有的正整数不超过 10^9。	首先通过调整法可以证明：将这 2nk 个数组成的序列递增排序之后，必然存在一个最优方案，其中每部分的两组内的最小值在排序后的序列中都相邻。 二分这个最大差异值 d，然后看所有排序后的序列中所有差值不超过 d 的相邻的数对。显然从左到右第 i（0<=i<n）个部分内的最小值在该序列中出现的位置不能后于 2ik。扫描一遍即知是否可行。	时间复杂度： $O(nklogMAXS)$ （其中 MAXS 为二分范围） 空间复杂度： $O(nk)$
2013 H	Матрёшка	给出一个正整数序列，要将它们合并为若干“套”，每套是一个从 1 开始的连续正整数序列。每次只能将两个连在一起的连续子序列进行合并，并且在合并时，除了某个子序列中小于另一个子序列内所有数的数不需要移动外，其它的数都要移动一次，求完成合并至少需要多少次移动，当然，也有可能无解。1<=序列长度 N<=500，出现的所有正整数均不超过 500。	预处理所有子序列内的最小值（$O(N^2)$）以及所有子序列里小于等于 1～N 内每个值的数的个数（$O(N^3)$），然后进行合并类 DP，根据这些预处理得到的信息可以保证 DP 在 $O(N^3)$ 时间内完成。	时间复杂度： $O(N^3)$ 空间复杂度： $O(N^3)$
2013 J	Pollution Solution	求一个以原点为圆心的半圆与一个完全位于 X 轴上方的简单多边形的相交部分的面积。3<=多边形点的数目 N<=100，所有坐标的绝对值不超过 1500。	首先根据多边形的有向三角剖分（在计算任意简单多边形面积时使用的），以原点为一个顶点将其剖分成 N 个带符号的三角形，然后问题转化为求一个以原点为一个顶点的三角形与一个以原点为圆心的半圆的相交部分的面积。根据三角形的另外两个顶点在半圆的内外分成三种情况（都在内、都在外、一内一外）分别处理，其中“都在外”的情况下，还要按照原点的对边是否与半圆相交分成两种情况处理。	时间复杂度： $O(N)$ 空间复杂度： $O(N)$

2013 K	Up a Tree	一个人写二叉树的前序、中序、后序遍历时，在递归过程里乱调用（比如中序遍历的过程中，可能调用的是左子树的前序与右子树的后序），已知三个过程里的输出位置都正确，且恰好有两个前序调用，两个中序调用与两个后序调用，但不知道它们的位置。现在给出这个人写的三个过程输出的某棵二叉树的前序、中序、后序遍历的结果，求出：（1）里面的六次调用分别是什么；（2）正确的前序、中序、后序遍历的结果。要求对（1）输出所有的解，对（2）输出字典序最小的解（先前序字典序最小，再中序字典序最小）。所有的前序、中序、后序序列均由大写字母组成，且在一个遍历序列里不会出现相同的字母（显然序列长度不超过 26）。	先枚举第（1）问的结果（共 90 种），然后记忆化搜索。注意任何时候问题都可以描述成：给出某棵二叉树的“前序”“中序”“后序”遍历的结果（加了引号表示是这个人写的过程输出的）中的若干个（至少一个），求出字典序最小的解。当三者都知道或者知道中序和另外两个之一时，解唯一，否则需要枚举左子树的大小。注意中间各种细节判断即可。	时间复杂度：$O(90\times N!)$（实际上根本达不到这种最坏情况） 空间复杂度：$O(N^7)$（其实也根本达不到）