泛做表格

姓名：岑宇阔

试题编号	名称	题目大意	算法讨论	时空复杂度
1998 E	Petri Net Simulation	给你 Petri 网的描述，包括 NP 个库所中的令牌个数，以及 NT 个变迁。让你模拟 NF 次变迁的发生。	模拟题。每次都暴力去寻找一个被允许的变迁，让其发生。	时间:O(NF*L) 空间:O(L)
1999 A	Bee Breeding	给定巢室节点的编号方法，对于每组询问(a，b)，输出 a 和 b 之间的最短距离。	先把正六边形巢室存在二维数组中。对于每组询问都做一次宽搜找最短距离。	时间:O(Max(a,b)) 空间:O(Max(a,b))
1999 C	A Dicey Problem	给定一张 R*C 的骰子地图。你可以通过转动水平或者竖直地移动骰子。如果当前骰子顶面的数字为 x，那么你只能将它移动到写着 x 的格子上，或者画着星星图案的格子上。求一条能回到起点的路径。数据保证最多只存在一条路径。	记忆化搜索。搜索中用 v[i][j][F][T]=0 或 1 分别表示骰子到达地图上的(i，j)位置，顶面为 F，面向我们的数字为 T 是否出现过。	时间:$O(6^2*R*C)$ 空间:O(R*C)
1999 E	Trade on Verweggistan	w 个工场，每个工场有 b 个放“prul”的箱子，箱子只能按顺序从顶到底购买，每个箱子有一个价格，箱子的卖出价都是 10 弗罗林，要你求出可以获得的最大利润。	DP。F[i]表示购买了 i 个箱子能获得的最大利润。对于每个工场，枚举买的箱子个数，然后更新 F 数组。	时间:$O(w^2*b^2)$ 空间:O(w*b)
2000 A	Abbott’s Revenge	给定一张箭头地图，每个路口用二维坐标表示。在每个路口处，如果你从某个方向进入了该路口，那么路口会有箭头告诉你下一步该怎么走。给定起点，终点的坐标，以及起始的方向，让你找到一条长度最短的从起点开始到终点的路径。	宽搜。F[x][y][d]表示从起点出发到达(x，y)，最后一步的方向是 d 的最短步数。	时间:$O(9^2*4^2)$ 空间:$O(9^2*4^2)$
2000 E	Internet Bandwidth	N 个机器，M 条无向边，边上有带宽(单位时间能传输的最大信息量)。求 S 到 T 的带宽。	裸的最大流。	时间:$O(n^2*m)$ 空间:O(n+m)
2000 F	Page Hopping	给你 n 个点和 m 条有向边，求两两之间的平均最短路径的长度。(1<=n<=100)	Floyd 求出两两之间的最短距离，然后取个平均。	时间:$O(n^3)$ 空间:$O(n^2)$
2001 A	Airport Configuration	有一个机场，到达的大门在北边，出发的大门在南边。每一个到达或出发的大门只对应一个城市，共有 n 个城市(n<25)。 你需要计算几个方案的客流指	模拟题。 对于每种方案，一一算出某两个门之间的客流指数，然后加起来。	时间:$O(n^2*m + m \log m)$ 空间:$O(n^2+m)$

		数。两个大门间的客流指数等于人数乘以距离，总的客流指数就是所有门之间的客流指数之和。		
2001 B	Say Cheese	在一块无限大的奶酪中，有 N 个球形小孔，小虫可以瞬间从某个小孔的一端飞到另一端。小虫能在 10S 咬穿 1mm 的奶酪。让你求出某个位置的小虫找到另一位置的小虫的最短时间。	把小虫当做半径为 0 的小孔。那么两点间的距离 F[i][j]=max(dist(i,j)-r[i]-r[j],0)。接下来就可以用最短路算法解决。由于 N<=100，可以用简单的 floyed 解决。	时间:$O(n^3)$ 空间:$O(n^2)$
2001 F	A Major Problem	西方音乐中，用 A 到 G 来表示 12 个音符，后面可能连有“#”或“b”。C/B# C#/Db D D#/Eb E/Fb F/E# F#/Gb G G#/Ab A A#/Bb B/Cb C/B# …上表中相邻音符构成半音，恰被一个音符隔开的两个音符构成一个全音。大调音阶由“全音-全音-半音-全音-全音-全音-半音”组成，且 A 到 G 每个字母恰好出现一次，第一个字母将额外在最后出现一次，同时音阶中不允许同时出现“#”和“b”。你需要写一个程序，在不同音阶中交换音符。	模拟题。暴力地把题目给的源调和目标调(如果合法的话)还原出来。然后回答询问。	时间:$O(7*2^6)$ 空间:O(24)
2001 I	A Vexing Problem	Vexed 游戏是一种类似于俄罗斯方块的游戏。游戏中，一面木墙上放着标有字母的石块，如果一个石块左边或右边是空的，那么它就可以移动一格。木墙上的格子不能移动。移动之后，如果某一个石块下面是空的，它就会掉下去。所有石块停止后，如果 2 个及以上标记相同的石块相邻，那么它们就消掉，同时形成多个组就一起消掉。消完了之后可能还有石块要掉落，然后再消掉…… 你需要给出一种用最少步数消光的方法。	搜索题。对于搜索时，我们可以用 Hash 去掉重复的状态。另外还有一个剪枝，如果一列上某种颜色只有 1 个，那么它必须要和左边或者右边的石块一起消，那么可以算出至少要移动几步。	时间:$O((NM)^{11})$ 空间:O(P) (P 是用来 Hash 的大质数)
2002 A	Ballons in a Box	已知一个长方体的盒子和一个 N(1<=N<=6)的点集，每个点代表一个可以放置气球的位置。在一个点上放置气球，就是以这个点为球心，让这个球膨胀，知道触及盒子的边缘或者一个之前已经被放置好的气球。你可以按某种顺序放置气球，要求气球所占据的总体积最大。	用 N!枚举气球放的顺序，然后按顺序计算每个气球的半径，然后求出最大的总体积。	时间:$O(N!*N^2)$ 空间:O(N)

2002 C	Crossing the Desert	你需要计算出从 1 走到 N 最少需要多少食物(1 到 N 都是二维坐标)。每走一英里，需要一单位的食物和水，食物只能在 1 购买，水可以在 1 到 N 收集。	最短路问题。如果我们知道了 y 到 N 的最少需要多少食物，那么通过 x、y 之间的距离，我们可以求得 x 经过 y 到 N 最少需要多少食物，然后我们可以用 Spfa 来做。	时间:$O(N^2)$ 空间:$O(N^2)$
2002 E	Island Hopping	有 n 个岛，给出它的路由器位置和居民数量。PIN 将会建一棵最小生成树，这样也许会有多种方案。所有的边都是同时开始建设的，每条边建完的时间跟边的长度成正比，当一座岛与 1 的连通时，这座岛就接入了互联网。 PIN 希望所有居民连入互联网的平均时间最短。	可以证明，任意的一棵最小生成树均可以保证平均时间最短。 然后只需用 prim 求出一棵最小生成树然后算出答案即可。	时间:$O(n^2)$ 空间:$O(n)$
2002 H	Silly Sort	有一个长度为 n 的需要排序的序列，在排序过程中，可以交换两个数的位置。每次交换需要一个代价，代价是被交换的两个数的和。 你需要写个程序计算把这个序列按升序排序的最小代价。	对于原序列的第 i 个数，找出它在按升序排序后的位置 p[i]，连一条边 i->p[i]，可以发现这张图由若干个环组成。 找出序列中的最小的元素 u。对于一个环，其最小的元素为 v，其和为 s，个数为 g。那么这个环的代价为 min(s+(g-2)*v，s+(g+1)*u+v)。	时间:$O(n)$ 空间:$O(n)$
2003 B	Light Bulbs	n 个灯泡，由 n 个开关控制。第 1 个开关控制 1、2 灯泡，第 n 个开关控制 n-1、n 灯泡，其余第 i 个开关控制 i-1、i、i+1 灯泡。要求你转换最少的开关使得灯泡状态 S 变为灯泡状态 T。	枚举第一个开关是否转换，然后可以依次知道其余开关是否转换，再判断第 n 个灯泡是否符合条件。	时间:$O(n^2)$ 空间:$O(n)$
2003 F	Combining Images	一种图像压缩算法是基于四分树编码的。进行四分树编码的图片必须是一个二进制像素方阵，方阵的边长必须是 2 的幂次。 如果一个图像中所有像素的颜色都相同，这个图像的四分树编码以 1 开始，然后是每个像素的颜色。 如果一个图像中含有不同颜色的像素，这个图像的四分树编码以 0 开始，然后依次是左上、右上、左下、右下的四分树编码。 然后把编码转换成 16 进制。 你需要写一个程序，计算两幅图	先把 16 进制编码转换成 2 进制编码。 对于两幅图像的编码，如果某一幅图像的编码长度只有 2（即 10 或 11）那么两幅图像的交很容易求得。 如果两幅图像的首位均为 0，递归处理左上、右上、左下、右下的 4 幅图像。如果 4 幅图像都交都是 10，那么这两幅图像的交也是 10，否则这两幅图像的交为 0 加上 4 幅图像的交。	时间:$O(Len^2)$ 空间:$O(Len)$ 其中 Len 为编码的长度。

		像的交。		
2003 G	A Linking Loader	给出 3 中语句 D、E、C。 D 表示定义一个符号的值，E 表示给一个符号编号，C 表示给一些数值分配地址。其中 C 语句中的$符号表示将后一个数字所代表编号的符号的值放入$符号指示的内存地址以及后面一个。 最后统计符号的值以及校验值。	考虑 D 语句存在向后引用的情况，用数组暂存内存的值，等到所有语句处理完后，对于每个符号的所有引用，修改内存中的某些值，最后再计算校验和。	时间:O(n*log S+M) 空间:O(n+S+M) 其中 n 是操作个数，S 是符号个数，M 表示内存单元的大小。
2003 H	A Spy in the Metro	地铁城有 N 个站台，列车双向运行，Maria 要在 T 时刻在 N 号站台会见当地间谍。请你帮她安排一个从 1 号站台到 N 号站台的方案，使得她在站台上有最少的等待时间。	记忆化搜索。搜索的状态是几号站台、方向、总的等待时间、总的时间，然后开个大的 bool 数组判断这个状态是否计算过。	时间:$O(N*M*T^2)$ 空间:$O(N*T^2)$
2003 I	The Solar System	已知开普勒三大定律，求一个行星在 t 时刻的坐标。行星轨道是以原点为中心的椭圆，右焦点是太阳。	算出行星的周期 T，把 t 对 T 取模，算出 t 时刻对应的面积 S1。二分一个角度，算出纵坐标，然后算出行星和太阳的线连扫过的面积 S2，比较 S1 和 S2。	时间:O(log(EPS)) 空间:O(1) 其中 EPS 是精度的倒数。
2003 J	Toll	你需要从 A 出发，到 B 卖出 X 把银匙。进入城镇，每携带 20 件货物需要交纳 1 件，进入村庄，需要交纳 1 件货物。（村庄编号为’a’-‘z’，城镇编号为’A’-‘Z’） 你需要算出在 A 需要携带几把银匙。	二分在 A 携带银匙的数目，然后用 dijkstra 算出到 B 最多能携带多少，跟需要卖出的 X 把银匙比较。	时间:$O(52^2 *\log Ans)$ 空间:$O(52^2)$
2004 E	Intersecting Dates	给定 NX 个已有日期区间，和 NR 个需要的日期区间。输出新增的日期区间。（日期区间从 1700 年 1 月 1 日到 2100 年 12 月 31 日，NX,NR<=100）	把 1700 年 1 月 1 日记为第 1 天，依次算出所有日期是第几天。把读入的日期转换成第几天，然后离散化，然后暴力寻找新增的日期区间，将其还原年月日。	时间:O((NX+NR)*NR) 空间:O(Len) 其中 Len 为日期的总长度。
2004 G	Navigation	题目让我们仿照 GPS 的导航，在二维笛卡尔平面内写一个简单导航的程序。题目规定了信号源的移动速度和信号的传播速度。	我们可以以信号源发出信号时的位置为圆心，发出信号时距接收点的距离为半径画圆，那么所求接收点的位置就是所有圆的公共点。然后先联立任意 2 个圆的方程，解出公共点，再代入其他圆的方程判断是否都成立。	时间:O(N) 空间:O(N)

2004 H	Tree-Lined Streets	地图上有 n 条大街，市议会想知道要种多少树。种树有 2 个限制:①一条大街上，每两棵树之间的距离至少 50m；②树与它所在的大街上十字路口的距离不少于 25m。	n 条大街，两两之间判交点。然后对于每条大街，对这条大街上的十字路口按 x 轴排序，依次枚举相邻的十字路口，计算种多少树，然后累加。	时间:$O(n^2 \log n)$ 空间:$O(n^2)$
2004 I	Suspense!	两幢距离为 d 的公寓的每层有猫、鸟或者什么也没有。每只猫只能往上跳小于 0.5m，或往下跳小于 3m。现要造一座水平的桥，使得猫不能通过桥抓到鸟，求桥缆最长是多少。	首先要求出桥面的最低高度。令 Hi 为第 i 层的窗台高度。通过观察 Hi-0.5 到 Hi+2.5，得出只要枚举 Hi-0.5，Hi，Hi+0.5 这 3 个高度找出最低高度就行了。 设抛物线为 $y=px^2$。Jan 家窗台为(X1，Y1)，Tereza 家窗台为(X2，Y2)。那么 p、X1、X2 都可以求出。然后用积分求出 $y=px^2$ 从 X1 到 X2 的长度。	时间:O(n+m) 空间:O(n+m)
2005 B	Simplified GSM Network	在二维笛卡尔坐标系中，有 b 个基站，手机总是连接到其最近的基站。一个基站覆盖的范围成为一个单元。 当一个开机的手机经过一个单元的边缘的时候必须切换到另一个基站。给出地图的描述组成，你必须计算出从一个城市到另一个城市的基站切换次数的最小值。	对于每条城市之间的连边，分治递归算出这条边基站切换次数。 然后 floyd 算出城市两两之间的距离，回答询问。	时间:$O(R*B*\log EPS+C^3)$ 空间:$O(B+C^2)$
2005 C	The Traveling Judges Problem	有 nj 个人，他们需要找到最便宜的租车方式使得每个人都到达目标城市。如果几个人在旅途的某一段坐同一辆租的车，就可以减少总费用。你的任务是找出这些人应该采取的路线使得总费用最小。	由于城市数 nc<=20，那么我们可以先枚举需要访问到哪些城市，然后不难发现接着应该求出最小生成树，这样总费用最小。	时间:$O(n^2 2^n)$ 空间:$O(n^2)$
2005 D	cNteSahruPfefrlefe	完美洗牌是将 52 张牌分成两半然后完美地交叉起来。0-51 完美洗牌的结果是 26 0 27 1…51 25。Preston 发现每次洗完牌后，他最多会把某两张相邻的牌互换。他希望你写一个程序来确定他的错误。(最多会洗 10 次)	定义两副牌的差异值为最少要交换多少次才能使得两副牌对应位置相等（可以交换不相邻的牌），差异值可以 O(52)得到。 先求出 0-51 完美洗牌 i 次的结果，然后和读入的牌组求差异值 diff，如果差异值 diff<=i，那么这副牌组就被洗了 i 次，且洗错了 diff 次。 然后从读入的牌组开始倒着洗牌去搜索错误的位置。(需要用可行性剪枝及最优性剪枝)	时间:$O(52^{Step})$ 空间:O(52) Step 为牌组被洗的次数。

2005 E	Lots of Sunlight	有 n 个公寓楼，从东至西排列，每栋楼有若干层，每层有一间公寓。太阳从东方升起，以恒定的角速度划过天空，然后从西方落下。当一间公寓的整块东侧或西侧外墙被太阳直射，或者处于太阳正下方时，我们就认为公寓受到太阳直射。求出 Q 个公寓的太阳直射时间。	对于每个询问，都向左向右求出最大的遮挡角度，然后算出遮挡的时间。	时间:O(Q*n) 空间:O(n)
2005 F	Crossing Streets	Peter 离开家步行去大学，他想知道怎样减少穿过街道的条数。 给定 n 条水平或竖直的街道以及家和大学的位置。（n<=500）	把 n 条街道以及家和学校的坐标离散化，然后从家的位置开始宽搜，算出至少需要穿过几条街道才能到达大学。	时间:O(n^2) 空间:O(n^2)
2005 G	Tiling the Plane	给你一个闭合的多边形，所有角均为直角，每条边的长度均为单位长度的整数倍，你可以随意复制这个多边形，也可以随意移动，但不能旋转或翻转。问这个多边形能否铺满整个平面。 只有 2 种本质不同的铺满平面的情况:使用正四边形铺满平面，或使用正六边形铺满平面。	把描述多边形的字符串(长度为 n)分成 2 个长度均为 n/2 的子串 s1，s2，枚举分割点 i[1],…,i[k](1=i[1]<i[2]<…<i[k]=n/2)，k=3 或 4。然后把 s2 的 i[j]到 i[j+1]的子串，j=1,..,k-1，分别翻转，比较 s1 是否等于 s2，如果相等，则找到了一种铺满平面的情形。	时间:O(n^3) 空间:O(n)
2005 H	The Great Wall Game	n*n 的网格上有 n 颗石子，一个格子中最多放一颗。每一次移动，可以将任意一颗石子移动到相邻的空方格中。你需要求出最少的移动步数，使得 n 颗石子排成一条水平、竖直或斜的直线。	首先枚举 2*n+2 条直线，对于给定的一条直线，求出每颗石子移动到直线上每个点的步数，然后用 KM 求出最小步数。（边权变负）	时间:O(n^4) 空间:O(n^2)
2005 I	Workshops	w 个会议，每个会议有人数和持续时间。r 个房间，每个房间有人数限制和最长持续时间。每个会议至多只能在一个房间举行，每个房间最多只能举行一个会议。要求最大化在房间举行的会议，如果有多个答案，要求最大化在房间的人数。	不难想到可以用贪心来做。w 个会议按人数不增排序，r 个房间也按人数不增排序。依次处理每个会议，对于每个会议，都去找一个满足条件并且最长持续时间最小的房间。这个可以用 multiset 来维护。	时间:O((n+m)log m) 空间:O(n+m)
2005 J	Zones	有 n(n<=20)个集合，以及集合之间的公共部分，共 m(m<=10)个。让你选择 k 个集合，使 k 个集合的元素最多。	先求出 n 个集合选 k 个的所有情况，对这些情况用容斥原理算出有多少个元素，然后取个最大的。	时间:O(C(n，k)*n*m) 空间:O(n*m+n^2)
2006 A	Low Cost Air Travel	有 NT 种机票，每种机票有一个飞行路线，你只能从路线的第一个城市出发，按次序飞，但你可以在中途结束这张机票的飞行。 给你 NI 条旅游路线，你要决定	F[i][j]表示经过了旅游路线的 1-i 个城市，现在在 j 城市的最小花费。然后 DP 即可。 注意，F[i][j] 也可以更新 f[i][*]。只要所有的 F[i][*]不再被	时间:O(NI*10*n^2) 空间:O(10n) 其中 n 为飞行路线的总城市数。

		怎样购买机票使旅行的花费尽量小。	更新，那么可以枚举 i+1。	
2006 B	Remember the A La Mode!	有 P 种饼干，I 种冰激凌，都有个数限制。每种饼干和冰激凌的组合都有一个收益，且每份组合都只需要一份饼干和一份冰激凌。 让你在所有饼干和冰激凌都用完的情况下，求出最小收益和最大收益。	每种饼干建一个点，每种冰激凌建一个点。S 向每种饼干连一条容量为这种饼干数量，费用为 0 的边；每种冰激凌向 T 连一条容量为这种冰激凌数量，费用为 0 的边；饼干与冰激凌之间连一条容量为 INF 费用为这种组合收益的边。在这张图中求最小费用最大流以及最大费用最大流。	时间:O(Flow*P*I) 空间:O(P*I) 其中 Flow 为最大的流量。
2006 D	Bipartite Numbers	二段数是这样的正整数：恰好包含两种不同的十进制数字 s 和 t，s 不为 0，并且 s 的所有出现均排列在所有 t 的前面，例如 44444411，还可以简洁地表达为 6 个 4 后面接 2 个 1。 给你一个 1-99999 的数字 n，要你求出比它大并且是它的倍数的最小二段数。	二段数可以表示为 LenA 个 A 减去 LenB 个 B（LenA>LenB，1<=A<=9，A-9<=B<=A，B≠0）。 先预处理出 Len 个 1 对数字 n 的余数，记为 f[Len]。通过 A*f[LenA]同余于 B*f[LenB]得知要维护一个 res[i][j] 表示 i*f[res[i][j]]同余于 j。 首先枚举 LenA，然后枚举 A，B，O（1）求得 LenB。	时间:O(Len*9^2) 空间:O(Len*9) 其中 Len 为所求二段数的最长长度。
2006 E	Bit Compressor	有一种压缩二进制串的方法是这样的：讲连续的 n 个 1 替换成 n 的二进制表示。(注：替换发生当且仅当这种替换减少了二进制串的总长度） 给出原串长 L，原串中 1 的个数 N，以及压缩后的串。(压缩后的串长度<=40 bits，L<=16 Kbytes）	搜索题。 对于压缩后的串中的各个 0，枚举它是不是原串的 0，然后统计答案，注意 2 个 1 是不能替换成$(10)_2$的。	时间:O(2^{Len}) 空间:O(Len) 其中 Len 为压缩后的串的长度。
2006 G	Pilgrimage	Jack 和朋友在旅行，由 Jack 来处理公共资产。有 4 种操作: 1.IN k，成员增加 k 个 2.OUT k，成员减少 k 个 3.COLLECT k，每人捐 k 元钱。 4.PAY k，支付 k 元钱。 成员人数增减的时候，Jack 会均分公共资产，但巧合的是，整个旅途中，均分公共资产的时候没有出现分数。求初始时有多少成员。	设初始公共资产为 M，初始人数为 P。再设 t 为平均资产，初始时 t=M/P，则 IN 和 OUT 不改变 t 的值，COLLECT k 会给 t 增加 k，PAY k 会给 t 减少 k/P。 可以知道在 IN 和 OUT 的时候 t 都是整数，而且 COLLECT 操作对结果没有影响。所以只要维护一下人数的增量 x，PAY 的和 y，在 IN 或 OUT 的时候，如果 y>0，那么 P+x 就整除 y，然后把 y 清为 0。要注意，y 的值要在第一次 IN 或 OUT 出现后再统计。	时间:O(n^2*k) 空间:O(n*k)

2006 I	Degrees of Separation	给你 n 个人，m 个关系。对于任意两个人，他们的分离度是联系两个人需要经过的最小的关系数。 让你求出网络中任意两人的分离度的最大值。	把人名用 map 容器转换成数字，然后 Floyd 即可。	时间:$O(n^3)$ 空间:$O(n^2)$
2006 J	Routing	给你 n 个节点，m 条有向边。你需要选出最少的节点，使得 1 可以通过沿着有向边到 2，并且 2 可以通过有向边到 1。（只能经过你选出的节点）（N<=100，M<=1000）	设 f[x][y]表示从 1 走到 x 和从 y 走到 1 这两条路径上的节点的最小值。 f[x][y] 可以更新 f[i][y]，f[x][i]，f[y][x]，然后 spfa 即可。	时间:O(K*N*M) 空间:O(K*N*N) 其中 K 是 spfa 的系数。
2007 A	Consanguine Calculations	有 ABO 和 Rh 两种血型系统。你会得到父母和孩子三者中两者的血型，请你确定剩下的那个人所有可能的血型。	暴力枚举剩下那个人的血型，然后题目就转换为已知三者的血型，判断是否合法。我们可以把两种血型系统分开处理。然后对于每种血型系统，我们可以手工算出所有情况，在程序中打表存储。	时间:O(1) 空间:O(1)
2007 I	Water Tanks	有 n 个水箱，从第二个水箱开始，每个水箱用一个管道与前一个水箱连接。连接水箱的管道是水平的，而且高度递增。水箱 1 是打开的，所以水和空气可以自由地从顶部流入，其他的水箱是封闭的。 你需要写一个程序来计算在水面到达水槽 1 的顶部前倒进水槽 1 的水的体积。	物理题。对于第 i 个水箱，先判断能否到达第 i 个管道的高度，不能的话，求出上升的高度。到达了第 i 个管道的高度，再判断第 i+1 个水箱能否到达第 i 个管道的高度，不能的话，求出上升的高度。第 i+1 个水箱到达了的话，再求出第 i 个水箱继续上升的高度。	时间:O(n) 空间:O(n)
2007 J	Tunnels	一个无向图，spy 要从 1 号点走到 0 号点，你可以随时观察到 spy 的逃跑路线并且任何时候都可以选择摧毁任何一条道路，使得 spy 无法逃跑到 0 号点。假设 spy 使用最优逃跑策略，问最坏情况下你最少需要摧毁几条道路。	设 f[i]表示 spy 从点 i 出发，使用最优策略逃跑最少需要摧毁几条道路。f[i]的初始值为源点为 i 汇点为 0 的无向图的最大流最小割。然后每次找一个最小的 f[i]，将 i 点删去，用 f[i]+剩下的图中 j 到 0 的最小割去更新 f[j]。	时间:$O(n^4 m)$ 空间:O(n+m)
2008 A	Air Conditioning Machinery	你需要在指定的空间内装一个管道。你只有 6 个弯管，每个弯管有 2 个口，占 4 个单位体积。你必须用最少的弯管连通入口和出口。	由于最多只能使用 6 个弯管，所以可以用搜索来做。	时间:$O(8^6)$ 空间:$O(\max(x,y,z)^3)$
2008 B	Always an Integer	给定自变量为 n 的多项式，问当 n 取任意正整数时这个多项式的值是否恒为整数。（最高次不超过 100）	记这个多项式的最高次为 m。可以证明，只需验证当 n=1 到 m+1 即可。	时间:$O(m^2)$ 空间:O(m)

2008 D	The Hare and the Hounds	猎狗和兔子在进行公路拉力赛，猎狗必须找到兔子选择的一条或多条路。在任何十字路口，他们都应该使用主要的道路规则，即转最小的角度。 在选择点，兔子可能违反主要道路规则，随机选一条路走。猎狗遇到选择点后，必须尝试每一条道路，直到到达一个确认标记。 在遇到一个不正确的路线后，猎狗必须原路返回，然后选其他路。	模拟题，按照要求一步步模拟即可。	时间:O(n*m) 空间:O(n+m)
2008 E	Huffman Codes	有 n 种字母，和每种字母的哈夫曼编码（根据字母的出现频率构建）。为简化问题，哈夫曼树的每个节点的左儿子的权值小于等于右儿子的权值，让你求出可能的频率分布种数。（n<=20）	根据哈夫曼树的构建和为简化问题的规定，我们可以得出哈夫曼树的所有节点的权值的大小关系（大于等于关系）。 然后根据这个大小关系来进行搜索。	时间:O(Ans*n) 空间:O(n) 其中 Ans 为可能的最大答案。
2008 F	Glenbow Museum	一个直角多边形是内角均为 90°或 270°的多边形。我们把 90°记作 R，270°记作 O，那么一个只包含 R 和 O 的角序列能够粗略地表示一个直角多边形。 一个角序列被接收只有满足存在一个地方能看到整个多边形。求长度为 L 的被接受的角序列的个数。	通过简单的内角和计算，我们得到 R 比 O 多 4 个。角序列被接收当且仅当角序列中不出现 2 个相邻的 O(首尾也算相邻)。 设 R 有 K 个，那么最终的答案是 C(K，4)+C(K-1，4)。	时间:O(1) 空间:O(1)
2008 G	Net Loss	给定一个最高次数为 n 的多项式。用 2 条线段(交点的横坐标为 c 且 c 介于-1 和+1)$y = a_1x + a_0$ 以及 $y = b_1x + b_0$组成的函数 g 来近似地代替原函数 p，要求函数 p 和其近似函数 g 的距离达到最小化。 定义函数 p 和 g 的距离为它们的差值的平方的定积分(-1<=x<=1)。	设交点纵坐标为 d。不难得到定积分是 d 的二次函数，不妨代入-0.5、0.5、1.5 来求出二次函数的系数。 当交点的纵坐标给定时，定积分左侧(-1<=x<=c)是 a_1 的二次函数，不妨代入-0.5、0.5、1.5 来求出二次函数的系数，然后返回最值，定积分右侧也同理。	时间:$O(n^2)$ 空间:O(n)
2008 H	Painter	给出 n 个三角形，先判断是否有三角形相交，若没有三角形相交则求被三角形包含次数最多的区域被包含次数+1。	首先随机角度旋转各个点的坐标，避免垂直线段的干扰。 接下来就沿 x 轴扫描，遇到线段左端点则把线段插入 set 中，并与它上方和下方的线段分别求是否相交。遇到线段右端点时把该线段从 set 删除，并判断原先在它上方和下方的线段是否相交。 此题需要特判掉来自同一三角形的情况，并且需要维护括号序列。	时间:O(n log n) 空间:O(n)

2008 J	The Sky is the Limit	有 n 座山脉，每座山脉看作是一个二维上的等腰三角形。一条天际线是一座或多座山脉的轮廓。让你求出天际线的总长度。	把三角形的二腰拆成 2 条线段。对于每条线段，跟其他所有线段判交，求出交点，设交点共 m 个，则把这条线段拆成 m+1 条线段。 最后把 x 轴所有出现过的坐标放入一个数组排序，对于每段小区间，分别计算长度。	时间:$O(n^4)$ 空间:$O(n^2)$
2009 A	A Careful Approach	有 n 段区间(n<=8)，让你在每个区间里选一个点，使得这些点之间的最小的间隔尽量大。	把这些区间进行全排列。 对于每一种排列，二分一个间隔，然后贪心判断这个间隔是否合法。	时间:O(n!*n*log Ans) 空间:O(n)
2009 D	Conduit Packing	给你 4 个圆的直径，让你求出最小的圆能够包含这 4 个圆，且圆与圆之间不能相交。	可以得出这样一个结论：4 个小圆的最优排列是按照顺时针顺序的。二分一个答案，枚举小圆的排列顺序，然后以大圆的圆心连向和 4 个小圆的切点，若 4 个小圆两两之间的角度总和小于或等于 360°，则合法。	时间:$O(4!*4^2*\log Ans)$ 空间:O(4)
2009 F	Deer-Proof Fence	有 N 棵树苗，M 表示围栏距树苗的最小距离。你需要设计若干个围栏，使得所有的树苗都被包围并且围栏距树苗至少为 M。围栏可能同时包括直线和曲线，你需要最小化围栏的总体长度。(N<=9)	w[S]表示集合 S 中的树苗被一个围栏包围的最小长度，可以由 S 组成的凸包的长度+一个圆的周长算得。 f[S]表示集合 S 中的树苗被若干个围栏包围的最小长度。 f[S] = min(f[S’] + w[S - S’])，S’为 S 的真子集。	时间:$O(3^n+2^n*n*\log n)$ 空间:$O(2^n)$
2009 G	House of Cards	Axel 和 Birgit 用 2M 张纸牌玩游戏。玩家交替进行操作，一步操作包括抽一张牌然后进行下列的一条(持有、形成基底、形成山峰)，如果组成三角形，那么分数会加到与 3 张牌多数颜色相等的那个玩家的分数上。 你需要读入一副牌和一个玩家的名字，算出那个玩家最多能赢多少(或者最少能输多少)。	题目展示的是一个双方博弈的游戏，两个人交替进行操作。由于 M<=13 比较小，不难想到用极大极小搜索加上 alpha-beta 剪枝来做。	时间:$O(2^{2M})$ 空间:O(M)
2009 H	The Ministers' Major Mess	n 个议案，m 个大臣，每个大臣会投 k(1<=k<=4)张票，你需要指出是否存在一种议案通过与否的方案，使得每个大臣都有大于一半的建议被满足。	当 k=1，2 时，大臣的建议都要被满足。 当 k=3，4 时，大臣的建议至多只有一个不满足，也就是说如果某一个不满足，那么剩下的 k-1 个都要被满足，可以用 2-sat 来做。	时间:$O(nmk^2)$ 空间:$O(mk^2)$

2009 I	Struts and Springs	一个最外面的窗口，n 个其内部的窗口，每个窗口都有一个直接包含它的的窗口。窗口有宽度和高度，以及该窗口的左上角相对最外面的窗口的坐标。除了最外面的窗口，每个窗口连着 6 根支杆或弹簧，1 根连接窗口的两条竖直边，1 根连接窗口的两条水平边，另外 4 根连接这个窗口的一条边与直接包含它的窗户的相应的边。弹簧会按比例压缩，支杆的长度不会变化。 m 个询问，当最外面的窗口的大小变化时，n 个窗口的位置和大小分别变成多少。	模拟题。对于每个窗口，先找出直接包含它的窗口。 然后对于每次询问，先修改最外面的窗口，接着修改被其直接包含的窗口，依次修改完所有的窗口。	时间:$O(n^2+m*n)$ 空间:$O(n)$
2009 K	Suffix-Replacement Grammars	后缀替换法由起始字符串 S 和一些后缀替换规则组成。每个规则是由 X->Y 的形式给出，其中 X 和 Y 是等长的由字母构成的字符串。这个规则表示如果你现在的字符串后缀是 X，你就可以用 Y 替代它。规则可以无限次使用。 你需要确定 S 是否能变成 T，如果可能，输出最小步数。	设 G(L)表示长度为 L 的字符串集合。D(L,X,Y)表示长度为 L 的字符串 X 转换成 Y 的最小步数，并且 X、Y 属于 G(L)。 G(L)只需包含 S、T 和规则前后的 2*NR+2 个字符串的长度为 L 的后缀（如果有的话）。 如果有规则 X->Y，那么 D(L,X,Y)=1。如果 X 和 Y 的首位相同，记 X0 为 X 去除首位所得的字符串，Y0 同理，那么 D(L,X,Y)=D(L-1,X0,Y0)。 对于同一个 L，D(L,X,Y)之间可以进行一次 floyd 求最小步数。	时间:$O(Len*NR^2)$ 空间:$O(NR^2+NR*Len)$ Len 为 S 的长度，NR 为规则数。
2010 B	Barcodes	Code-11 编码中的字符被限制为 0-9 和‘-’号以及特殊符号:开始和结束标志。 Code-11 编码会独立地编码每一个字符，由 5 个相邻的区域来编码。每个区域要么是宽的代表 1，要么是窄的代表 0。字符之间有 1 个窄的区域用来分割两个字符。 Code-11 编码时会用到两个检验字符 C 和 K，会根据前面的字符算出。	首先宽的区域的宽度是窄的区域的宽度的两倍，然后读入的区域宽度可能跟实际的宽度有 5%的误差。 我们先在读入的宽度里求出最小的 x1 和最大的 x2，然后≤105/95*x1 的都归为窄的，其它归为宽的，再检验宽的区域是否都≥95/105*x2，最后再判断是否存在一种情况，使得宽的是窄的两倍。	时间:$O(n)$ 空间:$O(n)$
2010 C	Tracking Bio-bots	m*n 网格的房间里，有 w 堵横墙。机器人每次只能向北或东移动，但不能通过墙。出口在东北角，问有多少个格子使得机器人不能到达出口。	由于 m、n 最大是 10^6，而 w 最大为 10^3，不妨先对坐标进行离散化，然后宽搜求解。	时间:$O(w^2+w \log w)$ 空间:$O(w^2)$

2010 D	Castles	给出一棵树，每个节点都是一座城堡，攻打这个城堡需要一定的士兵，会阵亡一定的士兵，还要驻守一定的士兵。 你可以任意选择第一个攻打的城堡，求最少需要的士兵数量。	首先枚举第一个攻打的城堡。 然后先攻打这个城堡，在递归处理各个子树算出攻打子树需要多少士兵，阵亡多少士兵(驻守可以当成阵亡)，然后按需要减去阵亡从大到小排序，依次攻打。	时间:$O(n^2 \log n)$ 空间:$O(n)$
2010 F	Contour Mapping	海拔信息的形式是一个整数序列，依次代表在一条从西往东的扫描线上取的等距离的点的海拔。扫描线间距的取值使得测量区域内所有不在边缘的每个测高点都有 6 个其他的测高点离他最近(正六边形的形状)。用线段连接每个点与距其最近的所有点，构造出许多三角形，并用三角形来估计实际地形。 需要你计算出所有高度是 h 的倍数的等高线的长度之和。	我们只需要统计每个三角形内部的等高线之和，和经过三角形边界的等高线之和。 前者可以用等差数列求出。后者就比较麻烦。对于一条边，如果其两个端点的海拔相等，并且是 h 的倍数，并且和这条边组成三角形的两个顶点的海拔都和这条边的两个端点海拔相等，那么这条边就不需要统计进去，否则要统计。	时间:$O(s*p)$ 空间:$O(s*p)$
2010 G	The Islands	二维平面上有 n 个点，按横坐标从小到大给出（横坐标两两不同）。船长每次从最西边的点（0 号点）出发寻找一条路径去最东边的点（n-1 号点），途中访问一些点，之后回来，途中访问剩下的点。每条路径船长必须一直往东或一直往西。 有 2 个特殊点 b1、b2，他必须在不同的路径中访问这两个点。计算用两条路径访问所有点的最短路径长度。	Dp。设 F[i][j]表示第一条路径为 i 到(n-1)，第二条路径为 j 到(n-1)并且 k=max(i,j)+1 到(n-1)都被访问过的最小路径长度。 不妨设 b1<b2(否则交换它们)，并且 b1 在第一条路径上，b2 在第二条路径上。 F[i][j]=min{F[i][k]+Dis(j,k),k!=b1, F[k][j]+Dis(i,k),k!=b2},k=max(i,j)+1。	时间:$O(n^2)$ 空间:$O(n^2)$
2010 J	Sharing Chocolate	给你一块 x*y 的巧克力，你可以沿着行或者列的分割线把巧克力分成两块，你可以将分成的小块继续分。 给你 n 个正整数，表示把 x*y 的巧克力分成 n 个部分各自需要包含的面积。	F[s][w]表示一块宽为 w 的巧克力能否分成集合 s 中包含的那些部分。 那么答案就是 F[2^n-1][x]。	时间:$O(3^n*Max(x,y))$ 空间:$O(2^n*Max(x,y))$
2010 K	Paperweight	一个纸镇由两个拥有一个公共面的四面体拼接而成。纸镇中有一粒芯片，我们需要计算出所有纸镇稳定地放置的方案中，芯片到纸镇底部的最小和最大距离。 如果在一种放置方案中，纸镇的重心向任意方向移动 0.2 个单位后，	先求出两个四面体的重心 G1 和 G2，纸镇的重心在 G1、G2 的连线上，且距离比等于质量的反比。(由于纸镇密度均匀，质量比=体积比，而体积可以由混合积算出) 接着枚举 3 个点构成一个平	时间:$O(1)$ 空间:$O(1)$

		纸镇仍不会发生移动，则这种放置方案被认为是稳定的。	面，判断另 2 个点是否在同一侧，如果在的话，算出重心在这个平面上的射影，判断是否稳定，稳定的话，算出芯片到这个面的距离，维护最小和最大值。	
2011 A	To Add or to Multiply	a-C-m 系列的指令集只有 2 种操作:+a，*m。 给你一个 a-C-m 处理器和 p、q、r、s 四个数。你需要构建最短的 a-C-m 程序，使得对于任意 p<=x<=q 返回一个输出 y 使得 r<=y<=s。	先枚举*m 的次数 g。 然后设第 i 次到第 i+1 次*m 之间有 h[i]个+a，则这个 a-C-m 程序即为函数 $F(x)=x*m^g + \sum h[i]*m^{g-i}$ (0<=i<=g)。 那么 r<=F[p]<=F[q]<=s，得到 $\sum h[i]*m^{g-i}$ (0<=i<=g)的范围，再得出∑h[i]的最小值。	时间:$O(\log^2(10^9))$ 空间:$O(\log(10^9))$
2011 E	Coffee Central	有一个面积为 dx*dy 的网格状的城市，城市中有 n 个咖啡馆。城市的两个交叉口(a,b)和(c,d)之间的距离为|a-c|+|b-d|。 有 q 个询问，每次读入一个正整数 m，让你确定一个最佳位置，使得距离最佳位置 m 以内的咖啡馆最多。	进行坐标变换。(x,y)->(x+y,x-y)。那么|a-c|+|b-d| = max{|a-c|，|b-d|}。把咖啡馆按横坐标排序。 对于每次询问，从小到大枚举横坐标 x，树状数组维护横坐标为 x-m 到 x+m 的咖啡馆的纵坐标，然后枚举可行的纵坐标 y，算出距这个点 m 以内的咖啡馆，更新答案。	时间:O(q*(n+S)*log L) 空间:O(n+L) S=dx*dy,L=max{dx,dy}.
2011 F	MachineWorks	有 N 台机器，对于每台机器，你知道其价格 Pi 和买入的时期 Di，然后从买入的第二天开始，每使用这台机器一天，就可以为公司创造出 Gi 美元的收益，你可以在买入之后的某一天，以 Ri 的价格卖出机器。 开始时公司拥有 C 美元，你需要在 D 天内买卖机器，由于空间的限制，公司在任何时间都只能最多拥有一台机器。求 D 天内公司可以得到的最大收入。	先对 N 台机器买入的时间 Di 从小到大排序。 设 Fi 表示到第 Di 天，卖掉之前的机器（如果有的话），最多能获得多少美元。 F[i]=max{C,max{D[i]*G[j]+U[j]}}，1<=j<i，并且 F[j]>=P[j]，其中 U[j] = F[j] + R[j] - P[j] - (D[j]+ 1)*G[j]。 我们可以对时间进行分治（cdq 分治），然后求出所有的 F[i]。	时间:O(N log N) 空间:O(N)
2011 H	Mining Your Own Business	给出一张图，选择一些点，使得图中删除任意一个点之后，剩余的任意一个连通块内包含一个被选择的点。求最少选择的点的个数以及最少情况下的方案数。	Tarjan 算法找出所有割点，对于剩余部分缩点，对于度数等于 1 的点，至少选择一个点，对于度数等于 0 的点，至少选择两个点。	时间:O(n+mlogn) 空间:O(n+m)

2011 I	Mummy Madness	给定 n 个木乃伊的坐标。初始时，你站在(0，0)，每一次，你可以移动到相邻 8 个格子之一(或者不动)，然后每个木乃伊都会移动到相邻 8 个格子之一，并且使得它与你的欧几里德最小。 求你最多能走几步。	不难发现可以二分答案。假设答案为 x，那么你可以到达的区域为一个边长为 2*x 的正方形，同理每个木乃伊可以到达的区域也是边长为 2*x 的正方形，然后要判断你的那个正方形的格点是否被木乃伊的正方形完全覆盖，这可以用扫描线来做。	时间:$O(n\log^2(10^6))$ 空间:$O(10^6)$
2011 J	Pyramids	有两类金字塔：一类是高金字塔石块数为 $1^2+2^2+3^2\ldots+n^2$; 另一类是矮金字塔 $1^2+3^2+5^2+\ldots+n^2$ 或者 $2^2+4^2+\ldots+n^2$ 。 给你用上 N 个石块搭若干个金字塔，要求金字塔数尽可能少，金字塔两两不同，金字塔至少包含 2 层。在满足上述条件的情况下，最大的金字塔尽可能大，最大的相同，满足次大的尽可能大……（N<=1000000）	可用的金字塔数不是特别大（最多 320 个）。 题目首先要求金字塔数尽可能少，不难想到用迭代加深来做。	时间:$O(Num^{Ans})$ 空间:O(Num) 其中 Num 为不同的金字塔的数量，Ans 为可能的最大答案。
2011 K	Trash Removal	给定一个 n 个点的多边形，你可以旋转它，使得它可以垂直落下通过一个宽度最小的管道。 求最小的宽度。	先求出 n 个点组成的凸包。不难发现下落时凸包的某一条边必须是竖直的。逆时针依次枚举这条边，会发现距这条边最远的点也是逆时针移动的，所以只需扫一遍即可。	时间:$O(n \log n)$ 空间:O(n)
2012 A	Asteroid Rangers	三维坐标上有 n 个点，每个点做匀速直线运动，求最小生成树改变的次数+1。	共 $O(n^2)$条边，任意两条边的大小关系最多交换 2 次，则可能使最小生成树方案发生变化的事件点最多 $O(n^4)$个。 对于每个事件点，定义在事件后长度变得更小的边为取代边，另一条为被取代边。如果被取代边不在当前的最小生成树上，那么最小生成树不会发生变化。只有在取代边的两端点在最小生成树的路径中包含被取代边，最小生成树才会改变。	时间:$O(n^4 \log n)$ 空间:$O(n^4)$
2012 B	Curvy Little Bottles	有一个由一条 X=Xlow 到 X=Xhigh 的多项式曲线绕 X 轴旋转一周构成的瓶子。瓶底在 X=Xlow 是实心圆，瓶口在 X=Xhigh 是敞开的。 你需要先算出瓶子的体积，然后你需要划定不超过 8 个的标记，标记	记多项式为 f(x)，那么所求瓶子的体积即为 ∫ $(PI*f^2(x))$ Xlow<=x<=Xhigh 的定积分。 标记可以用二分+定积分求得。	时间:$O(n^2 *\log EPS)$ 空间:O(n)

		间的体积是给定的（从瓶底开始划）。		
2012 C	Bus Tour	有 n 个位置，0 是总部，1 到 n-2 是旅馆，n-1 是景点。你需要找到一条路径，从总部出发，经过所有的旅馆，到景点，然后回来再一次经过所有的旅店，返回总部，要求去景点的路上接游客的前 n/2-1 个旅馆，在回来的路上也得是前 n/2-1 个让游客下车的，并使路途最短。注意公共汽车可能经过某个旅馆但是不停下来，之后再来这里让游客上下车。	F[0][i][X]表示从 0 出发经过集合 X，到达 i 最少需要多少时间，F[1][i][X]表示从 n-1 出发经过集合 X，到达 i 最少需要多少时间。F 数组可以用 DP(或记忆化搜索)求出。 枚举 n/2-1 个旅馆。枚举中间节点 i，算出从 0 出发，经过枚举的旅馆到 i，再从 i 出发，经过剩下的旅馆到 n-1 的最少时间 t1。同样的，回来的最少时间 t2，然后用 t1+t2 去更新答案。	时间:$O(n^2*2^n)$ 空间:$O(n*2^n)$
2012 D	Fibonacci Words	定义斐波那契 01 字符串。F(0)=0，F(1)=1， F(n)=F(n-1)+F(n-2),n>=2 。+是字符串的连接。 给定一个模式串 p 和一个数 n，求 p 在 F(n)中出现了几次。	先求出最小的 m 使得 F(m)的长度大于等于 p 的长度。记 s 为 F(m)，t 为 F(m+1)，那么 t+s 为 F(m+2)， t+s+t 为 F(m+3)， t+s+t+t+s 为 F(m+4)，不难发现相接处只有 t+s，s+t，t+t，并且之后都是 s+t 与 t+t 依次出现，那么只要算出 s 和 t 中出现次数以及 3 种相接处产生的额外次数（可以用 KMP)，然后简单递推即可。	时间:O(L + n) 空间:O(L) 其中 L 为模式串 p 的长度。
2012 E	infiltration	有 n 个单位，对于任意 2 个单位 A、B，要么 A 管理 B，要么 B 管理 A。 我们可以渗透到一个单位，那将使得我们控制该单位和被该单位直接管理的单位。 求渗透最少的单位使得控制所有单位。	题目相当于要我们求出一个支配集。 由于 n<=75，可以证明可能的答案最大为 5。那么我们就可以枚举所有的渗透情况，判断是否控制了所有单位。(本题需要用位运算来加速。)	时间:O(C(n,5)) 空间:$O(n^2)$
2012 G	Minimum Cost Flow	有一个从源点向汇点运水的系统，共 N 个点，M 条边。每个点有一个高度和这个点上洞的数量。你可以加入一条新的边连接 2 个洞，费用为两个节点的距离，购买塞子塞住一个洞的费用是 0.5。 你可以选择一个压力，把水注入源点，要求找到最小的费用使得水能流到汇点，同时保证水不会渗出洞。	首先 O(N)枚举水的高度。然后把小于等于这个高度的节点分成若干个连通块。 第 i 个连通块和第 j 个连通块之间的距离为第 i 个连通块内有洞的节点和第 j 个连通块内有洞的节点之间的最短距离。 然后求出 S 到 T 的最短距离。	时间:$O(M*N*\log N+N^3)$ 空间:$O(M+N^2)$

2012 K	Stacking Plates	有 n 堆盘子，每堆盘子从小到大堆成一堆。你需要将 n 堆盘子重新组合成从小到大的一堆。 你可以进行两种操作：拆分，可以将一个盘堆堆顶任意数目的盘子抬起，并放置在一侧，使堆一分为二；合并，可以将一个盘堆放置在另一堆的堆顶，前提是上方的堆最底层的盘子尺寸不大于在下方的堆最顶层的盘子尺寸。求最少的操作次数。	首先把 n 堆盘子放在一起排序，得出最终盘子的序列。 不然发现题意即要把这个序列连续地分割成若干块，使得每个块都来自同一堆，并使块数最少。如果盘子的大小各不相同，那么我们很容易得出答案。如果盘子的大小可以相同，那么需要DP，记 F[i]为结尾是第 i 堆的最少块数，每次把最终序列值相同的一段一起处理。	时间:$O(n^2*h)$ 空间:$O(n*h)$
2012 L	Takeover Wars	两家公司正在进行商业战争，分别是 X 和 Y。X 有 n 个子公司，Y 有 m 个子公司，你知道每个子公司的市场价值。 公司轮流行动（X 先动）。每次，该公司可以选择自己的两家子公司进行合并，产生的新的子公司的市场价值是参与交易双方的价值之和；也可以选择自己的一家子公司 i 和对方的一家比 i 价值小的子公司 j 进行吞并，吞并之后，i 的价值不变，j 从市场上消失。如果一个公司的子公司全部消失，那么它就输了。 问谁能赢得这次商业战争。	每次操作，每个公司要么用自己价值最大的子公司吞并对方价值最大的子公司（如果可以的话），要么合并自己价值最大的两个子公司。 然后你需要枚举第一次 X 公司选择吞并还是合并，接下来的最优操作都是确定的（优先吞并），只需模拟即可。	时间:$O(n+m)$ 空间:$O(n+m)$
2013 A	Self-Assembly	你需要写个程序来判断一个给定的 n 种分子的集合是否可能合成一个无限大的结构体。每个分子都在二维平面内被表示成正方形，四条边表示分子间连接的四个表面。每条边上的连接标识有 2 种，一种是一个大写字母加上一个“+”或“-”,另一种是“00”。两条边能并在一起当且仅当两者的字母相同，符号相反。(“00”不能和任意边并在一起)	对于一个四条边是 a、b、c、d(假设没有“00”)的分子，我们从任意一个出发，向另三个的相反符号连边。(即 a->b’，a->c’，a->d’，对 b、c、d 也相似处理)。这样连边的意义是:假设我现在要添加一个有 b 的分子，那么通过连 a->b’，我们可以转换为要添加一个有 a’的分子。最后在这张图中 dfs 判断是否存在环，如果存在，即为 unbounded。	时间:$O(52^3)$ 空间:$O(52^2)$
2013 B	Hey, Better Bettor	有一个赌场，每次赌 1 块钱，赢了返还 2 块。提供以下的优惠:你可以赌任意次。当你赌完以后，如果你的总资产减少了，这个赌场会把损失的 x%退还给你，但只能赎回一次。 给定 x 和赢的概率 p%，计算在最优策略下的最大期望收益。	由于 p<50，并且只能返还一次，那么实际上只能是赢了走人，输了退款。 赢到 b 块钱或者输到 a 块离场，那么赢钱的概率(设 $r=(p/(1-p)))P=(r^b-r^{b-a})/(1-r^{b-a})$。 那么三分枚举 a、b 即可。	时间:$O(\log^2(10^4))$ 空间:$O(1)$

2013 C	Surely You Congest	有一张 n 个路口 m 条双向道路的图，通过不同道路所需时间是不同的。所有的乘客都会从各自的路口出发，但所有乘客都会在同一个地点（路口 1）结束他们的旅程，你只能让他们走最短路。但乘客不能在相同的时间，从同一方向，开始沿相同的道路移动，问最多有多少乘客可以抵达路口 1.	从 1 开始求单源最短路，如果两个点到 1 的最短路不同，那么他们显然不会有冲突。 把乘客按照其所在路口到 1 的最短路长度排序。长度相等的乘客放在一起，在原图的最短路图中做一遍最大流，维护答案。（此题用 dinic 会 TLE，需要用普通的 bfs 增广求最大流）	时间:O(c*(n+m)) 空间:O(n+m+c)
2013 D	Factors	用 f(k)表示 k 的质因子排列数方案。给你一个 n，至少有一个 k 使得 f(k)=n，求最小的 k。（n，$k<=2^{63}$ ）	先求出 63 以内的素数(2,3,5…)。我们只需要算形如 $k=2^{p[1]}*3^{p[2]}*\ldots*61^{p[18]}$，且 p[i]>=p[i+1]的数，并把 f(k)算出，放入一个 map 容器中，每次询问只需要从 map 中查询。	时间:$O(num_k*63^2)$ 空间:$O(num_k)$ num_k 为不同的 k 的个数。
2013 E	Harvard	每一个访问数据的指令都由 2 个数控制。一个数 a。如果 a 为 0，那么访问的是 0 号内存库，如果 a 为 1 则访问 BSR 中选择的内存库。另一个数为 f 表示访问该内存库的第 f 个变量。假设每一个指令花费相同的时间运行，另外还有一个可以设定 BSR 值的命令。 一个程序是一个操作序列，有 2 种操作:1.变量访问，写作 Vi；2.循环，写作 Rn<…>E。 你需要算出一个程序最小运行时间。	先枚举哪些变量应该放在第 0 号内存库。 在程序中删去这些变量，再统计 i、j 前后相邻的次数 f[i][j]。然后搜索出剩下每个变量所在的内存库，根据 f[i][j]算出总操作次数。 注意，除了 0 号内存库，其它内存库是等价的，也就是在搜索剩下每个变量所在的内存库的时候要注意去重。	时间:$O(C(13,s)*L*s^2+B13)$ 空间:$O(L+n^2)$ L 为程序长度，n 为变量的个数，B13 是第 13 个贝尔数。
2013 F	Low Power	n 个机器，每个机器有 2 个芯片，每个芯片放 k 个电池。每个芯片能量是 k 个电池能量的最小值。现有 m=2*n*k 个电池，要把它们放在芯片上，使得所有机器 2 个芯片的能量之差的最大值最小。	把对电池的能量从小到大排序，那么每个机器的 2 个芯片的能量在排完序的数组中下标应该是连续的。 不难想到二分答案。假设现在二分的答案是 x，再设已经放好最小值的芯片的剩余位置有 y 个(初始为 0)。那么从小到大依次判断相邻位置，如果差≤x，那么把它们放入 1 个机器的 2 个芯片，把 y+=2*k-2。如果差＞x，并且 y 是 0，那么就不合法，否则把 y 减去 1。	时间:$O(mlgm+mlg10^9)$ 空间:O(m)

2013 H	Матрёшка	套娃是一些从外到里大小递减的传统玩偶组成。你只知道一个完好的套娃内的玩偶大小是从 1 到某个数字 m。 在组装套娃时，你只能将小的玩偶或套娃放入一个更大的玩偶中，并且每次只能把相邻的两个组合在一起。 你想尽快的组装好，唯一耗时的部分是打开一个玩偶并马上关上它。 求将 n 个玩偶重新拼成一些完好的套娃的最小代价。	用 g[i][j]表示区间[i,j]内的玩偶合并成一个套娃所需的最少操作次数(假如这个区间有相同的数，g[i][j]=∞)。 通过将区间[i,j]分成 2 部分[i,k]和[k+1,j]可以进行转移，主要的问题是计算最后一次合并的代价。 设 t 为[i,j]内最小的套娃的位置。当 k<t 时，k 倒着循环可以使得不需要拆开的玩偶个数单调不降。当 k>=t 时则正着循环。 有了 g[i][j]，剩下的问题就简单了。	时间:$O(n^3)$ 空间:$O(n^2)$
2013 I	Pirate Chest	有一个 m*n 的池塘，d(i,j)表示方格(i,j)的深度。你需要在池塘中放入一个长方体的宝箱，宝箱底面一边的尺寸不超过 a，另一边不超过 b，宝箱的顶面必须严格低于水面。 求宝箱的最大体积。	首先枚举宝箱底面一边所在的行 i，依次枚举宝箱底面该边的对边所在的行 j，维护出 f[k]表示 min{d(x，k)}，i<=x<=j。 然后用单调栈维护出 l[k]，r[k]表示当宝箱高度为 f[k]时，左右至多能延伸多少距离。然后通过(f[k]，l[k]，r[k])来更新答案。	时间:$O(m^2*n)$ 空间:$O(m*n)$
2013 J	Pollution Solution	给出一个在 x 轴上方，点数不超过 100 的多边形以及一个半径不超过 1000 的半圆，求多边形被半圆覆盖的面积。	对于多边形面积的求法，我们可以将问题简化为求一个三角形被半圆覆盖的面积，然后根据两个顶点表示的向量的叉积的值，将各个面积加减得到答案。	时间:O(n) 空间:O(n)
2013 K	Up a Tree	有 3 段求树的先中后序遍历的代码，三个过程中输出语句的位置是正确的，涉及到三个过程的六次递归调用中，恰好有两个调用先序，两个调用中序，两个调用后序，虽然可能在错误的过程中。 给出 3 段程序对于某棵树的先中后序，求出所有可能的代码，并且输出正确的先中后序。	先枚举 6 个递归调用分别是什么，然后用给出的三个字符串进行搜索，搜索的参数为错误的先中后序遍历程序的输出，由于递归调用可能是错的，这样搜索中可能要枚举根或者左子树的大小。 另外，还需要用 map 进行记忆化搜索。	时间:$O(n^8)$ 空间:$O(n^7)$