IOI2014 中国国家集训队第二轮作业解题报告

姓名：匡正非

试题编号	名称	题目大意	算法讨论	时空复杂度
2013				
2013 A	Self-Assembly	有 N 种小正方形，每种都有无限多个。每一种小正方形的四边都会有一个状态：-xx,+xx,或 00，xx 为大写字母 A-Z 中的任一个。每种小正方形都可以旋转或翻转。现在你可以将这些正方形拼接起来，要求是只有状态为+x 和-x 的边才能相接。00 不能与任何边相接。问是否有一种接法使得在正方形不重叠的情况下可以无限拼接。 N <= 40000。	因为每种小正方形都可以旋转或翻转，所以我们可以保持一直往上或往右拼接，因此正方形不能重叠的条件可以忽略。剩下的问题就是构出以每条小正方形的边为点的图，在图上找环的问题。但是 N<=40000，我们注意到边的种数只有 26*2+1 种，所以可以将原图缩成一个 O(26*2+1)的图。	时间： O（N） 空间： O（N）
2013 B	Hey,better bettor	现有一个有 p%概率赢的游戏，赢了可以得一元钱，输了则失去一元钱。当你输了钱时，你可以使用一次获得所输金额 x%的机会。已知 p <50，x <100，求最优策略下的期望得分。	因为 p<50，所以当我们使用了所给的一次机会后，我们便不会再玩下去。因为这个游戏非常简单，所以策略只关系到两个量:输到 L 时立即停止或赢到 W 时立即停止。我们设 P[x]表示玩了一阵之后，现有金额是 x，接着玩下去可以玩到 W 的概率是多少。 P[W] = 1, P[L] = 0. P[x] = p*P[x-1]+(1-p)*P[x+1]. P[x+1] = P[x]/(1-p)-P[x]*p/(1-p). 由特征根可以求得 P[x]的通项公式，而我们只关心的是 P[0]，即从一开始玩到胜利的概率。利用 P[0]即可算出对应一对 W,L 的期望得分 F[W,L]。打表可知，F[W,L]对于 W 和 L 都是单峰函数，所以用三分套三分之类的即可解决(或者求导)。	时间： O（log^2N） 空间： O（1）
2013 C	Sure Your Congest	有一个无向图，每一个节点都有一些人(可能为 0)。每个人都想通过最短的路径(可能不止一条)到达 1 号节点，但是如果两人同时在一个边上走的话，就会发生堵塞。问你在不堵塞的前提下，最多可以满足多少个人到达 1 号点。 N<=25000，M<=50000	若发生堵塞，则发生堵塞的两个人从各自起点到 1 号点的最短距离一定相等。 所以我们可以把到 1 号点距离相等的点分组来考虑。考虑网络流：对于一个到 1 点距离相等的点的集合 P，新建一个原点 S，令 1 为 T，如果一个节点有人且此节点属于 P，则从 S 向此处连一条边；对于原图中属于最短路的边都建一条同端点、流量为 1 的边，跑完最大流后统计答案即可。	时间： O(N*网络流复杂度) 空间： O（N+M）
2013 D	Factors	一个正整数(N)只有一种质因数分解的方法，但是将所有质因数展开后的排列可能有很多(F[N])。先给你一个正整数 K，求最小的 N 使得 F[N]=K。 N,K 均<2^63。	若 N = P1^e1*P2^e2*P3^e3...,其中 P1, P2..为质数且 P1<P2<P3..。用组合数可以算出 F[N] = sigma(e)!/e1!/e2!..。易知 e1>=e2>=e3 时 N 才可能对答案有贡献。所以可以暴枚 e1,e2,....，然后直接用 map 统计所对应的最小值即可。	时间： O（40000*logN） 空间： O(40000) （40000 为对答案有贡献的数的大概数量）
2013 E	Harvard	现有一种从多个内存库中询问数据的方法：这种方法分为两部分，一种是直接询问 0 号库，代价为 0；另一种是利用一个指针，将指针指向你需要询问的内存库（若已指向则不用在指）再询问，指针所指的内存库变化时将产生 1 的代价。给你内存库的个数 b，每个内存库的大小 s，一个包括询问，循环的程序，让你把所有被询问的数分到一些内存库中去，使得所有询问的代价和最小。输出最小代价。 变量最多只有 13 种。程序长度最多为 1000。	首先在 0 号内存库内的数据我们可以直接无视掉。 0 号库内的元素个数一定会达到最大。对于一个最优解，没有两个内存库中的元素合起来后仍能被一个内存库存起来。 所以这题就可以用暴搜。先枚举有哪些点属于 0 号存储点，接着用程序处理出点与点之间的关系（他们在程序之间相邻了多少次），然后再进行 DFS。	时间： 远小于 O(N*3^N) 空间： O(2^N*N)
2013 F	Low Power	给你一个 2*N*K（<=100000）的数组，你需要将所有的数分为 N 组，每组分为两小组，一小组 K 个数，同时令 D 为两个小组中的最小值的差。你的目标是使得所有 D 中的最大值 Dmax 最小，输出最小的 Dmax。	这题比较简单，可以直接二分 D，然后利用贪心的方法将序列扫一遍检验答案。	时间： O（NlogN） 空间： O(N)

2013 H	Matryoshka	有 N 个套娃排成一列，每个套娃都有一个权值。你每次可以将相邻两个套娃或者套娃集利用套娃的规则组合成一个套娃集，其中每打开并关闭一个套娃代价为 1。求将这 N 个套娃组合成一些没有缺失部分的套娃集(即不存在一段消失一段出现的情况)的最小代价。 N <= 500。	如果我们确定了最终所有的套娃集，那么这个问题就是一个 DP。 Dp[i] = min(Dp[j]+cost[j+1, i]) ， j < i 且 j+1 到 i 可以组合成一个完整的套娃。 问题是怎么求 Cost(i,j)。 我们发现，对于合并的最后一步来说，如果要合并的两部分已经确定的话，最后一步的代价一定是(i-j+1)-t，其中 t=max(x, 合并后的集合中最小的 x 个数均在合并前的同一个集合内)。所以我们可以推出式子： Cost(i,j) = min(Cost(i,k)+Cost(k+1,j)+(j-i+1)-t), i <= k < j。 直接计算 t 是 O（N^4）的，但可以优化到 O(N^3)。 利用 Cost(i,j)更新 dp[j]，最后 dp[N]即为答案。	时间 O（N^3） 空间： O（N^2）
2013 J	Pollution Solution	给定一个 N 条边的简单多边形，和一个圆心在(0,0)，半径为 R 的半圆，求简单多边形与半圆的交集面积。	一种方法是利用多边形的三角剖分，判交再统计面积。 这题因为精度不高所以还可以用 Simpson 做。 我用的是 Simpson。 注意：如果是用 Simpson，则需要强制递归到一定的层数，不然可能出现较大的误差。	时间： O(N*logN*Simpson 复杂度) 空间： O（N）
2012				
2012 A	Asteroid Rangers	一空间内有 N（N <= 50）个点，每个点在单位时间内都有一个运动矢量。运动时间不限，求这些点可能形成的 MST 个数。	对于任每两条边，求出它们长度相等的时刻(没有则忽略)，那么相邻时刻之间不可能产生新的 MST。对于每一个时刻，它对 MST 的意义就是所有边排序后某两条边的顺序交换了一下。又因为这两条边一条在 MST 中而另一条不再其中（否则不可能改变 MST）的可能性极小(大概最多 10000 次左右)，所以对每个可能改变 MST 的时候都直接 O（NlogN）求 MST 即可。	时间： O(10000*N*logN) 空间： O（N^2）
2012 B	Curvy Little Bottles	给你一个次数<=10 的函数 f(x)和三个正实数 inc,xmin,xmax，已知 xmin<=x<=xmax 时 f(x)>=0。求 f(x)从 xmin 到 xmax 的积分，以及所有满足 xmin<=x1<=xmax 且 f(x)从 xmin 到 x1 的积分=k*inc 的 x1，k 为正整数。	第一问：可以直接套用公式。 第二问：因为 f(x)从 xmin 到 x1 的积分单调递增，所以可以二分 x1。	时间： O(NlogΔx) 空间： O（N）
2012 C	Bus Tour	给你一个有 N 个点(N<=20)的简单图，每条边都有长度。你的任务是： 1：从 1 号点出发，浏览 2~N-1 号点（可以经过而不浏览），最后到达 N 号点。 2：从 N 号点出发，浏览 2~N-1 号点，最后返回 1 点。 此外，第一次浏览的前 H/2，(H=N-2)个点在第二次浏览时也必须先浏览，问最小的可行路径长度。	N<=20，自然想到状压 DP。 记 S[i][j]表示从 1 号点出发，现在在 i 点，状态为 j 时的最小花费； T[i][j]则表示从 N 号点出发的最小花费。 那么对于一个包含 1 的个数 Cnt==H/2 的状态 j，其对答案的贡献为： Min(S[x][j]+T[x][all^j]+T[y][j]+S[y][all^j])，2<=x,y<=N-1。	时间： O(2^N*(N+M)) 空间： O(2^N*N) 《》《》《》《》 Now at here
2012 D	Fibnacci Word	定义一种串 fib[i]，满足： fib[0] = “0”,fib[1] = “1”。 fib[i] = fib[i-1]+fib[i-2]。 其中的+表示将两串相接。 给你一个数 N(<=100)以及字符串 S，$\lvert S\rvert$<=100000，问 S 在 fib[N]中出现了多少次。	令 pre[i][j]表示 fib[i]长度为 j 的前缀是否能和 S 的某个后缀匹配，suf[i][j]类似，dp[i]表示 fib[i]中包含多少个 S。 我们可以利用 suf[i-1]和 pre[i-2]在 O($\lvert S\rvert$)内计算合并后产生的 S，再加上 dp[i-1]+dp[i-2]即为 dp[i]。但是 suf 和 pre 自身在$\lvert fib[i]\rvert$<$\lvert S\rvert$时不好转移，所以我们可以直接暴力求出第二个长度大于 S 的 fib[i]，复杂度为 O（$\lvert S\rvert$）。	时间： O（N*$\lvert S\rvert$） 空间： O（N*$\lvert S\rvert$）
2012 E	Infiltrastion	有一个 N 个节点(N<=75)的竞赛图，求这个图的最小覆盖集。	竞赛图的最小覆盖集是 O（logN）的，所以此题答案<=6，直接枚举是那些点即可。再加上压位即可过。	时间： O(C(75,6)*N/32) 空间： O(N*N)
2012 F	Keys	你有一些钥匙和金属环（均<=26），一些环是相连的，所有的钥匙都挂在环上。你有 4 种操作：把钥匙从环上取下来或装上去，把两个相连的环断开或两个断开的环接上。你的目的是将钥匙 A-M 与钥匙 N-Z 分开，并且两部分中的所有钥匙都要在一个由环组成的联通块内。求一种使得对钥匙操作最少，其次使得环操作最少的方案，输出两种操作的次数。保证金属环之间的连接不会产生环。	对于任一个环，如果其中 A-M（白色）的钥匙多一些，则一定会把此环中 N-Z 的钥匙（黑色）取走，反之亦然。而对于黑白两色数量相同的环，我们可以做一遍树 D 来确认最小的对环的操作。不过可能会出现所有的环都是白（黑）色的情况，所以还要枚举一个环，使得其无论如何都是一个黑（白）色的环。因为环数很小，所以复杂度不是问题。	时间： O（N^3） 空间： O（N）
2012 G	Minimum Cost Flow	有 N 个节点（<=400），节点之间可能有管道相连。给你起点 S 和终点 T，你的目标是从 S 输水到 T。有些节点破了一些洞从而会漏水，你有两种选择避免这种情况：一是在两个洞之间连一条管道，代价为两点间的距离；另一种是补洞，一个洞花费 0.5。你可以自定水位，求最小的花费。	显然我们关心的水位只和 N 个点的高度有关。于是我们可以从低到高不断加入点（等同于不断提高水位），然后将所得的子图利用贪心进行一些简单的转换，使得问题变为求 S 到 T 的最短路，SPFA 即可。	时间： O（N^3） 空间： O（N^2）

2012 I	A Safe Bet	一个 N*M 的矩阵(N,M<=1000000)，其中有 P 面呈'/'或'\'的镜子（P<=200000）。问有多少个位置，使得你在这里添置一面镜子后，从起点(1,0)向右射出的一条光线会从(N,M+1)向右射出。	显然答案就是从起点向右射出后被折射行成的光线与从终点向左射出行成光线的所有交界处。用树状数组和扫描线可以轻松搞定。	时间： O（NlogN） 空间： O（N）
2012 K	Stacking Plates	有 N 堆盘子(N<=50)，每堆盘子有 Hi(Hi<=50)个盘子，而每个盘子都有其大小，每堆盘子都是从小到大叠放的。 现有两个操作：把一堆盘子分成两堆，和把两堆盘子合成一堆（前提是一堆盘子里的所有盘子都不大于另一堆中的盘子）。问最少要多少步骤使得所有盘子都堆成一堆。	因为合并次数=分裂次数+Hi-1，所以我们只要最小化分裂次数即可。如果每个盘子大小都不一样，我们可以把所有盘子按大小 sort 一遍，然后只要求出相邻两项不在一个堆中的对数即可。而盘子有相同大小时，可以直接用 DP 求解。	时间： O（N^3） 空间： O（N^2）
2012 L	Takeover Wars	A 和 B 两个人分别有 N 和 M 堆石子(1<=N,M<=100000)，两人轮流进行这样的操作：把对方比自己最大的石堆小的某个石堆取走，或将自己的某两堆石子合并。 当一个人没有石子时对方胜利。问先手是否有必胜策略。	显然最优策略是这样的： 1.每次合并自己最大的两个石堆。 2.吞掉自己能吞掉的最大的石堆。 当一个人的对手行动结束后，对方最大的石堆仍比他的小，那么这个人必胜（不断吞对方的石堆）。 所以只需枚举先手第一步的策略是 1,2 中的哪一个，然后两边不断合并（吞掉还不如合并，想一想）直到出现上述的情况判断即可。	时间： O（NlogN） 空间： O（N）
2011				
2011 A	To add or to multiply	有一种程序有一个变量 X 与两个指令： A：将 X += a； M：将 X *= m。 现给你 a,m,p,q,s,t，求一个最短其次字典序最小的程序，使得 X 取 p-q 之间时所得的值在 s-t 之间。 1<=a,m,p,q,s,t<=10^9。	枚举用多少次 M 指令，判断是否合理，然后利用贪心在 M 与 M 之间从左到右插入 A 指令即可。可以证明字典序最小和长度最小能同时保证。	时间： O（log^2S） 空间： O（logS） 其中 S=10^9。
2011 B	Affine Mess	给你三个初始点，三个目标点，问三个初始点是否能在经过旋转，放缩，平移之后变成三个目标点。所有点都视作是相同的，且旋转操作必须放在最前。可行的旋转角度已给定，<100 种。	枚举旋转角度和点与点之间的对应关系，再利用代数直接判断是否可以通过平移，放缩达到目的。	时间： O（3!*100） 空间： O（1）
2011 C	Ancient Message	有六种符号（具体见题面）和一个 01 矩阵(<=200*200)，找出矩阵中出现过的所有符号（细节可能不同）。任两个符号不会相交，且所有符号都是四联通的。	初看此题很难，仔细观察后却发现每个符号包含的联通块数量不一样，所以直接 BFS 判断即可。	时间： O（N*N） 空间： O（N*N）
2011 E	Coffee Central	给你一个 R*C 的图（1<=R,C<=1000），其中有 N 个点（N<=5*10^5）。现有 Q 组询问（Q<=20），每组询问中有一个值 M（M<=1000000），让你求一个点，使得上述 N 个点中满足到这个点曼哈顿距离<=M 的数量最多。	将坐标转换可将问题变为矩形求交，直接扫描线就可过。	时间： O(Q(N+RC)) 空间： O(RC)
2011 F	Machine Works	由 N 件物品，每件都只能在 Di 天用 Pi 元买到。一天只能持有一种物品，且持有的物品可以创造 Gi(Gi>0)的收益。你可以在任意一天将某物品卖掉，获得 Ri（Pi>Ri）元，买卖可以放在一天，但是买卖物品的那天无法获得物品带来的收益。你一共有 D 天可以规划，并可以在 D+1 天卖掉你保留的物品，求最大收益。 N <=10^5, D <=10^9。	我们令 Ci+x*Gi 表示在第 x 天持有物品 i 时的最大收益，其中 Ci 为常数。那么问题就是求 Ci。显然，求 Ci 等价于求到 Di-1 天时的最大收益。转移满足单调性，所以看起来可以使用斜率优化 DP 求解。 但是结果是否定的，因为 Gi 不一定随 Di 递增。 所以可以利用动态凸包或者是 CDQ 分治之类的东西。	时间： O（NlogN） 空间： O（N）
2011 G	Magic sticks	有 N(N<=500)条线段首尾相连，形成一条很长的棍子。你可以在棍子中分成一些连续的部分，并一一拼成多边形，问多边形面积和最大为多少。	如果只有一个多边形，那么当答案最大时，显然所有的节点都在一个圆上。圆的半径可以通过二分求出，注意有两种可能的情况：圆心在多边形中和不再其中。 而当有多个多边形的时候，原问题实际变成了一个 DP。直接 DP 是 O(N^3)的，不过我们发现：当一个多边形面积比将其分开后各面积和小的时候，罪魁祸首一定就是其中最长的边，所以直接将这条边删去再分别递归即可，O(N^2)内即可求解。	时间： O(N^2) 空间： O(N)
2011 H	Mining your own business	一个有 N 个点（N<=50000）的矿洞，让你在最少的点上凿逃生井，使得任意一个点坍塌后除这个点外的所有点都可以不经过此点而到达某个逃生井。问最小凿井数和方案数。	对于图中的每一个双连通分量，只要在其中任一个点（除了割点）凿井，就可以保证其中的每一个点都能到达该井，或者通过某个割点到达其他的逃生井。除此之外，如果这个联通分量可以通过不同的割点到达其他的两个井，那么其中的所有点也可以保证。所以先求出双连通分量，对于只有一个割点的分量，直接在其中放一个井；否则不管它。特殊情况：如果一开始只有一个双连通分量，则这里要凿两个井。	时间： O（N+M） 空间： O（N+M）
2011 I	Mummy madness	你在一个很大的矩阵上的一个方格内，周围有 N 个（N<=100000）木乃伊。每次你可以向周围走一格，然后所有的木乃伊都会向你的方向也走一格，当你走到有木乃伊的格子内，或木乃伊走到了你的格子内，你的脑子就会被吃掉。。。。。问你能坚持的最长时间。所有生物所在的坐标绝对值均<=10^6。	可以证明如果所有木乃伊的移动范围(正方形)的交集可以覆盖你的移动范围时，你一定会被吃掉，否则一定不会。所以对时间做一次二分，扫描线加线段树判断即可。	时间： O（Nlog^2N） 空间： O（NlogN）

2011 J	Pyramid	有两种金字塔：一种是 1*1+2*2+。。。的，另一种是 1*1+3*3+5*5。。。的。现在你有一些金字塔，其方块数之和等于 M(M<=1000000)，求这些金字塔可能的构成方案，若有多种，输出第一个最大（相等时第二个最大，以此类推）的方案。	这题相当于一个背包，可能用到的金字塔可能只有 400 个，所以直接暴力做即可，不过要加点优化。 另外应为这题的答案不会超过 6，所以也可以爆搜（多加点剪枝）。	时间： O（400*M） 空间： O（M）
2011 K	Trash Removal	给你一个多边形（点数 N<=100），现要用两条平行直线把这个多边形夹住，问这两条直线之间的距离最小为多少。	求出该多边形的凸包，因为只有一条直线与凸包的某条边重合时答案才可能最小，所以枚举即可。	时间： O（N^2） 空间： O（N）
		2010		
2010 B	Barcodes	一些互不相同的五位二进制码分别代表了 1-9,-,开始/结束标志(*)（详见原题）。已知一段编码（长度为 N）开头和结尾都是*，中间的符号都是 1-9 或-，倒数第二，三位则分别由前几位递推过来（详见原题）。编码是用宽和窄的线条描绘的，宽线条正好是窄线条的两倍，且两个符号中间会用一个窄线条隔开。不幸的是，线条的宽窄可能会有 5%的误差。问这段编码是否能被解读，且倒数 2,3 个符号是否是正确的。 注意：标准的线条宽度可能是实数，但打印的线条宽度是整数；你可以将编码倒着来解码；打印宽度<=250，编码长度<=150。	以精度为 0.01 枚举窄线条长度，直接模拟判断即可。	时间： O（N*25000） 空间： O（N）
2010 C	Tracking Bio-bots	有一个 N*M 的矩阵（1<=N,M<=10^6），矩阵上面有 w 堵水平的墙(w<=1000)。问有多少个空白的格子只需往右、上走即可到达格子(N, M)。	离散化，然后暴力 BFS。	时间： O（w^2） 空间： O（w^2）
2010 D	Castles	有 N 个（N <= 100）城堡，城堡之间有一些双向道路，任两城堡之间都有且只有一条路径互相通达。你现在要派兵攻打所有的城堡，每个城堡都有一个占领值和花费值，即占领所需要的兵力和损失的兵力。你可以从任意一个点作为根开始攻打，问你最少需要带多少兵（军队可以从子树向根折返）。	N 很小，所以可以枚举根。那么问题就只需要决定子树遍历的顺序了。设 Ci 为该节点及其子树的损失总值，Pi 为总的占领值，设一共有 rest 个人来分配。对于两颗子树 a 和 b，当 a 必须在 b 前面时，可以发现： Rest > Ca, Rest > Cb Rest-Ca >= Pb, Rest-Cb < Pa Pb+Ca <= Rest < Cb+Pa Pa-Ca > Pb-Cb 因此把所有子树按 Pi-Ci 排序即可。	时间： O（N^2logN） 空间： O（N）
2010 F	Contour Mapping	有 N 个节点(N<=10000)，节点的分布规律详见原题（太难表示了- -，不过这是一个平面图 and 弦图 and 由很多正三角形组成）。已知每个节点的高度值(<=10^6)和一个数 H(H<=1000)，问你所有高度为 x*H(x 为非负整数)的等高线长度之和。	暴力+公式算出每个三角形内所有等高线的长度和，再暴力算出所有可能在三角形边上的等高线即可。代码比较繁琐，情况比较多，比较难调，所以这是一道耐心题。	时间： O(N*10^6/H) 空间： O(N)
2010 G	The Islands	平面上有 N 个点（N<=100），每个点的 x 坐标互不相等，两点的距离即为他们的欧几里得距离。 你现在要从 1 号点走到 N 号点（路径 1），再从 N 号点返回 1 号点（路径 2），两条路径处 S，T 外不能相交，且要覆盖所有的点。 此外，有两个特殊的点，这两个点不能出现在同一路径上。求最短的路径。	我们可以把路径 2 也看作是从点 1 到点 N 的路径，那么可以用 DP 解决：设 DP[i][j]表示路径 1 走到了 i，路径 2 走到了 j 时的最小距离，O(N)转移即可。	时间： O（N^3） 空间： O（N^2） 注：这题其实我 WA 了。
2010 I	Robots on Ice	有一个 N*M 的矩阵(2<=N,M<=8)，你的目标是从点(1,1)不重复地经过所有点到达点(1,2)。有 3 个特殊的点，第 i 个点需要你在 floor(i*N*M/4)步时恰好经过。问你有多少种方案。	此题无法插头 DP，所以暴搜无疑。 考虑加以下剪枝： 1.任何时候未走的点必须联通。 2.判断是否要判断上述情况。 3.任何时候目前所在的点到某一个特殊点的哈密顿距离必须不大于到此点的剩余步数。 4.不能提前到任何特殊点。	时间： O(?)(暴搜) 空间： O(N*M)
2010 J	Sharing Chocolate	你有一块 X*Y(1<=X,Y<=100)的巧克力，你需要把它切成很多个小矩形分给 N(N<=15)个人。每个人对分得的块数都有自己的要求，且你每次只能整列或整行地切。问你是否每个人都能分到想要的块数。	设 Dp[i][j][S]表示还剩 i 行 j 列，剩余的人状态为 S 时是否可行，可以通过枚举 S 的子集转移。直接这样转移太慢，但是因为 S 确定时剩下巧克力的块数也确定了，所以可以把 j 给去掉。最后就是卡常数了~	时间： O(3^N*NX) 空间： O(2^N*X)
2010 K	PaperWeight	有一个由含有全等的一面的两个四面体（保证面积均>0）拼成的均匀的 6 面体，其中有一个特殊点 p。现让你在 6 面中选一些面作为底面，使得这个 6 面体的重心往任意方向移动 0.2 都不会使得此物体倾斜。求出 p 到所有可行的面的最小，最大距离。	直接求出重心，然后对所有面暴力判断是否可行即可了。 就当是立体几何练手题。 （有一个特例：如果 6 面体有 4 个顶点在同一面上，那么实际这两个面起作用的部分是这四个点的凸包而不是仅仅两个三角形。）	时间： O(1) 空间： O(1)
		2009		
2009 A	A Careful Ap- -proach	有 N(N<=8)架航班，其起飞的时间在[ai,bi]之间(0<=ai<=bi<=1440)。问怎样安排航班，可以使任两个航班的起飞时间间隔的最小值最大，输出这个值。	N<=8，所以可以暴枚飞行顺序。接着二分加贪心判断即可。 注意最小间隔可能是实数，尽管输出要求是整数。	时间： O(N!logT) 空间： O(N)
2009 B	My Bad	给你一个有 N(N<=8)个输入，G(G<20)个点，U(U<20)个输出的逻辑电路。现这个逻辑电路中可能出现了一个（或多个）错误，如果只有一个错误请你将其找出，否则输出无法判断（或没有错误）。	因为只用判断唯一的一个错误，所以枚举错误，模拟运行一遍电路，在与所给的输出对比一下就行了。	时间： O(N^2) 空间： O(N) 这里的 N<20。

2009 C	The return of Carl	一个正八面体和其表面上的两个点，求这两个点在正八面体表面上的距离。	显然，我们可以把正八面体按照某种方式展开，然后求得两点在平面内的距离。直接展开复杂度太高，但是我们发现有很多的情况都是不合法的，所以把所有可能合法的方案构造出来再计算就行了。	时间：O(8!) 空间：O(1) 实际时间复杂度很小。
2009 D	Conduit Paking	有四个已知大小的圆，相对位置随意。你需要用一个圆把所有的圆包含在里面，且这 4 个圆两两不能重叠。问这个圆半径最小是多少。	我们可以用贪心证明，每个圆一定会和构造的圆以及另外的一个圆相切。因此二分圆的半径，枚举放圆的顺序，然后判断就行了。	时间：O(4!logN) 空间：O(4)
2009 E	Fare and Balanced	有一个 N(N<=10000)个点的拓扑图，图中只有一个起点和终点。你需要改变一些边的权值，使得从起点到终点的所有路径长度相等，且每条路径只经过不超过一条改变过了的边。问起点到终点的距离最小是多少（或者输出无解）。	一种显然的算法是：对于一个点，求出它到终点的最大距离，然后把它连出去的所有边都按照这个距离依次检查修改一遍。 但是这样有个问题：一个点在修改后到终点的距离并不一定等于它在修改前到终点的最大距离。仔细观察可以发现，有一个东西遗漏了：如果一个点必须被改变，那么从起点连向它的所有路径都不能再改变。因此做两边 BFS 即可。	时间：O(N+M) 空间：O(N+M)
2009 F	Deer-Proof Fence	有 N(N<=9)个点，你要用一些围栏(不必联通)将它们围起来，使得每个点到围栏的距离不小于 M。问围栏长度的最小值。	对于一个点的集合，围住他们所需的围栏长度实际就是这些点的凸包向外扩展 M 后的周长。因为 N 很小，可以用状压 DP 来枚举点集，然后求凸包计算就行了。	时间：O(N^2*2^N) 空间：O(2^N+N)
2009 H	The Ministers' Major Mess	有 N(N<=100)个议案，M(M<=500)个大臣，每个大臣都会有 si(si<=4)个提议，每种提议都是对某个议案的否决或同意。问是否有一种议案通过方案，使得每个大臣的超过半数的提议都能实现。	因为一个大臣最多只会有一个提议被否决（如果没有问题更简单），那么他的任两个提议的否决一定是不能同时存在的，因此这问题可以转换成一个 2-SAT 问题。	时间：O(N^3) 空间：O(N+M)
2009 I	Struts and Springs	有 N(N<=500)个窗户，任两窗户的边界不会相交。现有一个很大的窗户，左上角处在(0,0)，所有的窗户都被其包含。每一个窗户都会与六根弹簧或直杆相连，这六根分别连接了窗户的两条水平边，两条垂直边，以及这个窗户和直接包含它的窗户的四条相对应的边。当一个窗户大小发生变化时，他直接包含的窗户也会发生变化，其中弹簧会按比例伸缩，而直杆长度不会变化（链接中不可能全是直杆）。现有 Q(Q<=500)组询问，每次告诉你最大的窗户的长和宽变为了多少，请你计算出所有窗户左上角的位置，长以及宽。	模拟就行了。。。（这题就是卡题意的）	时间：O(NQ) 空间：O(N)
2009 J	Subway Timing	一棵含有 N(N<=100)个点的树，每条边都有一个权值 S(S<=60)。你可以将一些 S 变为 S-60，使得树中路径的权值和的绝对值的最大值最小。	答案是不大于 120 的，可以通过调整法证明。 首先二分答案，然后可以发现：对于 i 的子树中的所有路径的权值，如果最大值确定，那么其最小值在两者相加的绝对值不大于答案的时候必然是越大越好。因此设 dp[i][j]表示所有从 i 的子树连向 i 的路径中的最大值为 j 时最小值最大为多少，然后就是经典的树 D 了。	时间：O(NS^2logS) 空间：O(NS)
2009 K	Suffix-Replacement Grammers	给定一个起始串 S，一个目标串 T 以及一些变化规则 P，一个变化规则可以看做两个长度相同的串(A,B)：当一个串的某个后缀为 A 时，可以用 1 点代价将这个后缀转换为 B。问从 S 转换到 T 的最小代价为多少。S，T 长度相等，每个串长度都<=20，且规则数 N<=100。	我们观察任两个串 A,B（长度均为 len），可以发现，当它们满足： 1.有一套转换可以把 A 转移到 B。 2.有一套转换可以把 A.substr(1~len)转移到 B.substr(1~len)，且 A[0]=B[0]。 这两个条件中的任一个，则可视为这两个串之间可以转移。 我们考虑用 Floyd 算法，每次选取转换中的某个点，用上面的方法判断是否可以通过 t 来转移再用松弛操作更新。	时间：O(N^3) 空间：O(N^2)
		2008		
2008 A	Air Conditioning Machinery	有一个 X*Y*Z 的立方体(X,Y,Z <= 20)和一个出口，入口(均为单位立方体一个面)。你需要用不超过 6 个类似于俄罗斯方块中的'L'(可以改变方向)的管道把入口和出口连起来，问最少要用多少个'L'。	因为只有不超过 6 个，直接暴搜妥妥的。	时间：O(8^6) 空间：O(X*Y*Z)
2008 B	Always An Interger	给你一个次数在 100 以内的关于 x 的多项式，验证此多项式对于任意 x 是否都等于另一个给定数 D(<=10^9)的倍数。	此题有两个方法： 一个我们知道一个 E 次的多项式最多有 E 个解（无限解除外），而在剩余系内该条也是成立的，所以直接枚举 x=0~E 验证即可。 第二个我们可以利用差分。f(x)能被 D 整除的充要条件就是 f(0)被 D 整除以及 g(x)=f(x)-f(x-1)对于任意 x 均能被 D 整除。易知 g(x)次数一定比 f(x)小 1，所以不断递归下去就行。	时间：O(E^2) 空间：O(E)

2008 E	Huffman Codes	给你一颗二叉树(叶子节点个数 N<=20)，问你有多少种方法将大小为 100 的权分配给树中的叶子节点(顺序不同算一种)，使得所有叶子节点按照哈夫曼码的编码方式可以形成所给的树。	没有什么多项式的算法，所以只能暴搜加剪枝了——我们可以把权值从根节点向下分配。可以加这么些优化： 1.当一个节点分配权值给两个儿子时，两儿子的权值都不能比之前所有节点分配时产生的权值大。 2.我们发现优化 1 会因为分配的顺序而产生问题。但是我们发现，因为 1 中的规则，我们只要从当前仍未分裂的所有节点中挑取权值最大一个开始分裂即可找出所有的方案而不重复了。	时间： O(?)(暴搜) 空间： O(N)
2008 F	Glembow Museum	现一个给定的数 N(<=1000)和有两种转角：顺时针转 90°(L)或 270°(R)。你要从一个起点出发，往右走任意距离，转一个角，这样转了 N 个角后回到原点(到达原点后和右方向的夹角也要算)。你的路线不能相交，且路线围成的图形中，有一个点可以用直线看到图形的所有区域。问有多少种转角方法。	题目其实等价于求包含 N/2+2 个 L，N/2-2 个 R，且任两个 R 不能在一起的的序列个数。直接用组合数求。	时间： O(N) 空间： O(N)
2008 H	Painter	一个平面中有 N(<=100000)个三角形，你要判断是否有两个三角形相交而不包含。如果有，输出‘ERROR’，否则输出所有三角形被包含次数中的最大值。	提到被包含，就不能不提扫描线，此题的算法就是扫描线。扫描线做法大致是这样的：用一条和 y 轴平行的直线从左至右扫过，用 Splay 维护目前被扫的所有线段的上下关系(如果没有相交这个关系是不会变化的)。每次扫到一个三角形时，查找在它上面的第一条线段所属的三角形，这个三角形必然包含原三角形，或是与原三角形同级。但是此题会有线段相交的关系，那么我们直接在一条线段加入 Splay 时与和他 compare 过的所有线段判断是否相交就行了。	时间： O(NlogN) 空间： O(N)
2008 I	Password Suspects	给你一个数 len(<=25)和 N(<=10)个串，让你求有多少种串包含了 N 个给定的串。	建出 AC 自动机，记 dp[i][j][S]表示长度为 i，AC 自动机状态为 j，已匹配的串状态为 S 的串有多少即可。	令 M 为 AC 自动机节点数。 时间： O(Len*M*1024*26) 空间： O(Len*M*1024)
2008 J	The Sky Is The Limit	有 N(<=100)座左右对称的山峰，求所有山峰并集的上半部分轮廓线长度（不包括 X 轴)。	可以看出，轮廓线实际可以看做许多段线段，每一段都是当前最高山峰的一部分。 所以可以将山峰之间的所有交点求出来，离散化处理。虽然情况比较多。	时间： O(N^2) 空间： O(N^2)
2008 K	Steam Rollar	有一个 N*M(N,M<=100)的网格和一个起点，终点——很明显你要从起点到终点。通过网格边需要时间，而你在经过一条边时只要处于启动，结束，准备转弯或者刚刚转完弯的状态时，所花时间都会翻倍(最多翻一次)。求最短路。	还是老办法 SPFA（or Dijkstra)，不过要多记一些状态，比如上条边是否已翻倍，是否上个经过的点是拐点等。	时间： O(NM*logN) 空间： O(N*M) 当然了，常数比较大。
		2007		
2007 A	Consanguine Calculations	现有一个家庭(父，母，子)，已知父母和儿子中的两个人的血型(ABO 以及 Rh)，求剩下那个人的所有可能血型。	直接暴枚，轻松愉快。	时间： O(?)(暴枚) 空间： O(1)
2007 C	Grand Prix	现有一个山坡，坡面可以看做一个笛卡尔坐标系(x 轴即为山坡上升的方向)。已知山坡的角度 theta(0<=theta<90)以及一些坡面上的平面向量。第一个向量从原点出发，第二个从第一个的终点出发，以此类推。问至少要将所有向量沿原点旋转多少度，使得所有的向量起点的高度都不大于终点高度。	我们会发现 theta 好像没什么用，所以只要保证所有向量旋转后的Δx>=0 就行了，这个实际就是一个区间求交的问题。 不过阴险的 theta 不会这么放过你的。。。Theta=0 的时候要特判。。	时间： O(N) 空间： O(N)
2007 E	Collecting Luggage	有一个简单多边形，一个行李箱会从多边形上的一个点出发，然后按顺时针沿其移动，速度为 v1。另有一人在多边形外的一点，行走速度为 v2(v2>v1)。人不能穿过多边形，问他最早多久可以拿到行李箱。	因为 v2>v1，易证明时间与是否能拿到包的关系是有单调性的。 所以二分答案，在原多边形的点和此刻包所在的点之间走最短路即可。 有一个问题，就是有可能多边形的两点连线在多边形内部，而和其他边不相交，这个用中点判一下就行了。	时间： O(logT*N^2) 空间： O(N^2)
2007 F	Marble Game	有一个 N*N(N<=4)的平板，上面有 M(>0)个球和洞，每个球都有一个 1-M 的编号（互不相同），洞也一样。球和与其编号相同的洞一一对应。现你可以把平板按四个方向倾斜，使得所有球都到其对应的洞里（注意一个洞只能放一个球)，求最少的倾斜次数。	还是暴力，用 string 记状态，map 做数组然后写个记忆化搜索吧。	令状态数为 K。 时间： O(K*M*4)(暴力) 空间： O(K*M)
2007 G	Network	现有 N(<=5)个信息要发送，它们被分成片段后组合成了 M(<=1000)个信息包。你需要把所有信息连续地（即不能被打断）输送过去。传送信息的顺序任意，但信息包的顺序是确定的，所以你有一个缓冲区，用来缓冲一些片段并选择适当的时间发送出去。求一种方案，使缓冲区包含信息量的最大值的最小。	N 很小，所以枚举信息被发送的顺序，贪心模拟就行了。。。	时间： O(N!M) 空间： O(M)

<table>
<tr><td>2007 I</td><td>Water Tanks</td><td>有 N(<=10)个圆柱容器(底面积=1)从左到右放置，且只有最左边的圆柱上端开口。第 i(1<=i<N)个圆柱到第 i+1 个圆柱会有一个很小的管道，高度为 h[i]，h[i]根据 i 递增。问当第一个圆柱注满水时，一共注入了多少水。(大气压也要考虑，1m 深的水水压=0.0097 大气压，且密闭空间的气体满足 p1v1*p2v2)。</td><td>注意单从水压来讲，我们直接可以用一号点的高度减去所求点的高度即可，而算上气压后实际就相当于解一个一元二次方程。一路算过去就行了。。</td><td>时间：
O(N)
空间：
O(N)</td></tr>
<tr><td>2007 J</td><td>Tunnels</td><td>一个 N(<=50)个点，M(<=1000)条边的无向图，其中 1 号点是起点，N 号点是终点。你要阻止一个人从一号点到达 N 号点，方法就是删掉一些边。你可以随时观察到这个人在哪个节点，但是当他在边上时你不能将这条边删掉。问最小要删多少条边。</td><td>方法是这样的：在间谍行动的时候不断删掉一些边使得他不能到一些点，最后一次性解决问题。我们发现如果让间谍从一个当前花费最小的点到花费更大的点，肯定是不值的。所以可以一轮轮删除花费最小的一系列点，同时更新其他点所需的花费(最小花费加上这个点在剩余图中的最小割)，一直这样做直到只剩一个点即得答案。</td><td>时间：
O(N^3*M)
空间：
O(M)</td></tr>
<tr><td colspan="5">2006</td></tr>
<tr><td>2006 A</td><td>Low Cost Air Travel</td><td>有 N(<=20)个城市，M(<=20)条航线，每条航线都有一个价格，而你只能从航线的起点出发，在任意一个中转站结束。给你 Q(<=20)对点 S、T，问从 S 到 T 的最小花费。</td><td>对每组询问单独做，设 dp[i][j]表示在城市 i，目前搭的航班是 j 时的最小花费。用最短路跑就行了。</td><td>令 P=N*M
时间：
O(Q*PlogP)
空间：
O(N*M)</td></tr>
<tr><td>2006 B</td><td>Remember the A La Mode</td><td>有 P(<=100)种饼，I(<=100)种冰淇淋，每种饼和冰淇淋都有限，而两两之间都有一个匹配值。已知饼数总和等于冰淇淋总和，你要将所有饼和冰淇淋一一匹配，使得匹配值总和最大。</td><td>典型的最小费用最大流模型。</td><td>时间：
O(P*(IP))
空间：
O(IP)</td></tr>
<tr><td>2006 D</td><td>Bipartite Numbers</td><td>一个二段数是一种类似于 444000,777444 的数，可以用{i,j,s,t}表示(i 个 s 和 j 个 t)。其中左边一段的数不能是 0。
现给你一个数 N(N<10000)，问最小的二段数 D 使得 D 是 N 的倍数且 D!=N。</td><td>令 A[i] = 10^i, B[i] = {1111..}(i 个 1)。易知 D:{i,j,s,t}是 N 的倍数当且仅当：(B[i]*A[j]*s+B[j]*t)%N=0。
依次枚举 i+j，i，s，t 即可(i+j 大概 10000 之内)。一旦找出了满足的数，就不用在枚举更高位数的 D 了。再加一些优化即可过掉。</td><td>时间：
O(10000^2*81)(实际很小)
空间：
O(10000)</td></tr>
<tr><td>2006 E</td><td>Bit Compressor</td><td>规定一种二进制的压缩方法：把所有连续且极长的 1 压缩成一个表示这些 1 的个数的二进制数（前提是这样可以使串长变短）。如 111111110010011111111111111110011 会被压缩成 10000010011110011。
先给你一个压缩后的串 T(|T|<=40)问是否有多个原串 S(|S|<=16000)可以压缩成 T。</td><td>|T|<=40，所以可以暴力枚举。
我们发现，新串中所有的 1 都肯定是压缩的部分，除非是 1,10 这样的情况。而所有压缩的部分必然是 1 开头，结尾的后一位和开头的前一位也必然是 0。
所以可以从左到右找到所有需要压缩的满足前一位是 0 的 1，然后枚举这段压缩的长度，一直做下去即可。</td><td>时间：
O(?)（暴搜）
空间：
O(|T|)</td></tr>
<tr><td>2006 G</td><td>Pilgrimage</td><td>现有一个团体，有 T 个操作，分为三种：
IN x：x 人加入了团体。
OUT x：x 人离开了团体。
PAY x：团体的人 AA，一共出了 x 圆钱。PAY 的钱最多为 2000。
注意每人出的钱都是整数且完全相等。（我至今没完全读懂题意，看了别人标程才知道 PAY 是可以合并的且在所有 IN/OUT 之前或之后的 PAY 不算）
团体在任何时候都有至少一个人，问开始时可能的人数为多少。若有无限种，输出人数至少为多少。</td><td>一种想法是枚举所有可能的人数然后判断，但是这样太慢了。
我们可以利用第一次 PAY 时 x 的所有因数作为可能的值，然后一一判断即可。
当然在没有 PAY 时就有无限种可能了——通过 IN 和 OUT 来判断人数最小值。</td><td>时间：
O(sqrt(2000)*T)
空间：
O(T)</td></tr>
<tr><td>2006 I</td><td>Degrees of Separation</td><td>有一个 N(<=50)个点 M(<=2500)条边的无向图，问图中任两点的最短路的最大值。</td><td>Floyd。</td><td>时间：
O(N^3)
空间：
O(N^2)</td></tr>
<tr><td>2006 J</td><td>Roating</td><td>有一个 N(<=100)个点 M(<=1000)的有向图，给你一个起点 S 和终点 T，求一条 S 到 T 再从 T 到 S 的路径，使得两路径的并集点数最少。</td><td>这是一道很巧妙的题。令 S-T 的路径为 L1，再把 T-S 的路径反过来，看做是另一条 S-T 的路径 L2。设 dp[i][j]表示 L1 已到 i 点，L2 已到 j 点时的最小费用。一种普通的想法是让 L1 或 L2 走一步，然后更新。
但是如果只看 dp[i][j]中的 i，j 是否相同是会有漏洞的。可以发现，当两个点在 X、Y(X!=Y)，走了一些步后又成了 Y、X 时，X、Y 还是有可能会被算成两次。但是此时 L1 和 L2 都只会按 X 到 Y 的最短路走，所以用最短路更新就行了。</td><td>时间：
O(N^3)
空间：
O(N^2)</td></tr>
<tr><td colspan="5">2005</td></tr>
<tr><td>2005 B</td><td>Simplified GSM Network</td><td>在一个平面上，有 B(<=50)座塔，C(<=50)个城市，城市之间有 R(<=2500)条道路。显然平面上每个点都会有一个塔离其最近，而对于一座塔，这样的点可以形成一个区域。一条路的费用就是经过他时切换区域的次数。现给定一起点，终点，求从起点到终点的最小费用。</td><td>题目所述的实际就是一个 V 图，因为范围较小，可以用半平面交暴力构造出来。因为 V 图的每个区域都必然是一个凸多边形，所以对于每条边，我们只要判点和凸多边形是否相交即可。最后 floyd。</td><td>时间：
O(N^2logN+N^3+MN)
空间：
O(N^2)</td></tr>
<tr><td>2005 C</td><td>The Traveling Judges Problem</td><td>有一个 N(<=20)个点，M(<=400)条边的无向图，其中一些点有人。每个人都想去 1 号点，所以请你设计一个方案，使得所有人到一号点的路径并集的长度和最小。</td><td>看起来像最小生成树，实则不然——有些点我们不需要连接。枚举一下要用的点，然后再跑最小生成树即可。</td><td>时间：
O(M*2^N)
空间：
O(M)</td></tr>
</table>

2005 E	Lots of Sunlight	有 N(N<=100)栋公寓，每栋公寓都有 Xi 层房间。 所有房间的宽均为 W，高均为 H。 每天阳光都会于 5:37 从东边出来，在天空中匀速划过，然后于 18:17 在西边降落。每一个房间能被照射当且仅当它的西或东墙被阳光完全直射，或者太阳在它的正上方。 给定 Q(<=1000)个房间，计算这些房间被直射的时间段。	首先这个时间段一定是连续的，所以我们只用找出开始直射的时间和结束的时间。 显然这两个时间就是这个房间的东(西)墙的下端点和其他公寓的顶端所连直线中斜率最大的两条所代表的时间，直接枚举即可。	时间： O(QN) 空间： O(N)
2005 I	Workshops	有 N(<=1000)场会议，M(<=1000)个房间，每个房间只能容纳一场会议，且这场会议的人数和时间都要不大于房间的容量和可用时间，求一种安排方案，使得不能被容纳的会议最少，其次这些会议的总人数最少。	我们可以按可用时间将房间排序，再扫一遍：对于一个房间，显然在剩余会议中，选择他能容纳的人数最多的会议是最优的，否则无论他是在之后的房间内，还是不在任何房间内，都可以用调整法调整成这样。	时间： O(NlogN) 空间： O(N)
2005 J	Zones	有 N(<=20)个集合，每个集合都有一个大小。这些集合会形成 M(<=10)个交集，其大小和包含它的集合都已知。现要你从中选 K 个集合，使得其并集大小最大。	N<=20，直接暴枚即可。	时间： O(C(N,K)*M*N) 空间： O(N+M)
2004				
2004 C	Image is Everything	有一个 N*N*N 的大立方体，其中的一些单位立方体已经缺失（剩下的部分不一定联通）。每一个剩余的单位立方体六面都会被涂上同样的颜色，且重量均为 1。现在给你这个大立方体六个面的视图，求其最大重量。	实际上这问题等价于在这个立方体上不断找矛盾并删除一些不可能的方块，没有了矛盾的话，这时的重量就是最大重量了(具体证明可以看我的第一轮作业)。	时间： O(N^3-N^6) 空间： O(N^3)
2004 E	Intersecting Dates	现有一条线段，你有 M(<=100)个区间需要查询，而之前已经查询了 N(<=100)个区间。请你将所有未被查询的区间输出。	把所有区间端点离散化，然后从前往后扫一遍即可。	时间： O(M+N) 空间： O(N+M)
2004 H	Tree_Lined Streets	在一个平面中有 N(<=100)条街道，一些街道会有交点（下称十字路口）你可以在街道上植树(和现实差异确实比较大)，但要满足以下条件： 1.任两棵树的距离不小于 50M。注意这个 50m 不是直线距离，而是在街上的距离。 2.任一棵树到所有十字路口的距离都不小于 25M。 问最多可以种多少树。	由题易知，不同街上或者在一个路口两端的树之间必然不会有影响，所以对于一条街，我们可以处理出在他上面的所有十字路口，接着通过这些路口和它的两个端点将此街分段，然后每段分开处理就行了。	时间： O(N^2) 空间： O(N)
2004 J	Air Traffic Control	没看懂原题。。。不过这题就是给你 N 个(<=100)点（飞机）和 M 条线段(<=100)，每条线段包含一个 cnt。然后对于每条边，你要选择一个飞机，使得这条线段的两端点和飞机行成的外接圆中正好包含 cnt 个飞机，并将这 cnt 个飞机的权值加 1。如果找不到，输出 Impossible，否则输出最终所有飞机的权值。（感谢 hza 提供题意）	直接模拟。	时间： O(N^3) 空间： O(N)
Rest				
2003 H	A Spy in the Metro	有一条包含 N(<=50)个车站的地铁线，车站从左至右为标号为 1~N，相邻两车站间有一个行驶距离。一个间谍从 1 号车站出发，要求要在 T(<=200)时刻到达 N 号站。已知有 M1 辆列车从 1 号点出发到 N 号点，又有 M2 辆从 N 号点到 1 号点，它们的出发时间都已知。间谍可以在一些车站停留，但停留时间越少越好。请你计算这个停留时间的最小值。	设 dp[i][j]表示当前在 i 号车站，时间为 j 时的停留时间最少为多少，然后直接 DP 即可。	时间： O(N*(T+M1+M2)) 空间： O(N*T)
2003 J	Toll	有一个 N=52 的有向图，你需要把 K(<=1000)个物品从给定起点点运到给定终点。这 52 个点中有 26 个点是城镇，过去时每运了 20 件物品要交一件(不足 20 也算)，26 个点是乡村，过去只需交一件物品。问最少要从 1 号点运走多少物品？	显然答案具有单调性，可以二分。 二分之后令 dp[i]为在 i 号点时剩余的物品最多为多少，跑一边最短路就行了。	令 M 为边数，则 时间： O(logK*(N+M)logM) 时间： O(M)
2002 A	Balloons in a Box	有 N(<=6)个点和一个大小为 X*Y*Z 的盒子，你要按某种顺序依次放下这些点，放下后这个点会不断膨胀（球），直到他碰到了之前的点膨胀后行成的球或是盒子。盒子外的点不能膨胀。求一种顺序，使得所有点膨胀后的总体积最大。	N<=6，暴枚顺序模拟即可。	时间： O(N!*N^2) 空间： O(N)
2002 H	Silly Sort	有一个长度为 N(<=1000)的序列，你需要交换一些点的顺序，使得此序列变为一个升序序列。每次交换的代价是两个点的和，求代价总和的最小值。	我们可以把原序列看做一个置换，并把它分解成很多小置换。对于一个小置换，当它的大小为 1 时，显然不需要变动，否则所有点都肯定要离开自己最初的位置。如果要变动，我们可以从这个或者其他的小置换中选择最小值来将其一一复原，两种方案的 min 就是该置换还原为升序的代价。将所有代价累加就是答案。	时间： O(N^2) 空间： O(N)

1998 A	Crystal clear	有一个无限大的平面，平面上的每一个整数点都是一个在平面上的直径为 1 的圆的圆心。现在给你一个平面中的简单 N(<=25)边形，求被这个多边形完整包含或仅被多边形从圆心切割的所有圆在多边形内的面积和。	有两种做法： 第一种是把简单多边形扩展成一个矩阵(xmin~xmax, ymin~ymax)，然后对于这个矩阵内的所有点都计算一下。 这个做法比较暴力，不忍心的同学可以试试另一种：我们把每条边往周围扩展一下，然后对于所有扩展后的图形内的点都计算一下。 具体操作见我的第一轮作业吧。	令 M 为多边形坐标的范围。 时间： O(M^2*N) 空间： O(N)