张恒捷

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试题编号 | 名称 | 题目大意 | 算法讨论 | 时空复杂度 |
| 2001B | Say Cheese | 三维空间中有很多球，球可以重叠。在球内走的速度可以视为无穷大（即不花时间瞬间到达）。否则速度为1。现在要从一个点走到另一个点，求最快时间。  球数n<=100 | 将n个球心与起点终点搞出来，I到j的边权值为max(0,球心i与球心j的距离减球i半径减球j半径)  然后跑最短路。 | 时间：  空间： |
| 2003A | Building Bridges | 一张n\*m的格子图，图上会有一些建筑。你可以在图的分割线上建桥，桥必须是笔直的，并且恰好连接2座建筑。桥可以交叉。求把所有建筑都连起来的最短的桥长度总和。  N,m<=50 | 每座建筑去找沿格子线最近的建筑，然后把这些边全记下来，做最小生成树。 | 时间：  空间： |
| 2003B | Light Bulbs | N个灯泡和开关排成一排，每个开关可以控制与其距离不超过1的灯泡（第i只控制(i-1),I,(i+1)这3只灯泡），第1，n只开关稍有不同。开关一按，所有它控制的灯泡状态相反。现在要把这些灯泡从某种状态变到全部熄灭，求最少操作次数的方案。次数相同比较字典序。  状态都要用10进制读入/输出。 | 如果第一只开关的状态知道了，后边的开关有没有按都能推出来。 | 时间：  空间： |
| 2003C | Riding the Bus | 一个这样分形的图案，分形1次的图案是左图，分形时每次都把原图复制9遍，然后用长为len的线接起来，如右图。  在分形n次的图案中，每len的距离就有一个关键点。现在给出两个点A,B，离A，B最近的关键点为C,D。现在A要走到C，再走到D，再走到B。C到D要沿着黑线走。求距离。  N<=8 | C-D的距离就是（0,0）到D的距离-（0,0）到C的距离。  Work(n,x,y,f)就是在分形n次下，要求(0,0)到(x,y)的黑线距离，f是’s’型图案是否翻转。每次递归到分形n-1的情况下去做。 | 时间：  空间： |
| 2003D | Eurodiffusion | 有n个国家，每个国家是平面上的一个矩形，每个整点上都有一个城市。每个城市初始时都有100W的次国硬币。每天每个城市回向相邻的城市送出的所有硬币。求每个国家所有城市都拥有至少一个其他所有国家硬币的时刻。  N<=20, 平面<=10\*10 | 答案不会超过10\*10\*1000，枚举时间模拟一下。 | 时间：  空间： |
| 2003E | Covering Whole Holes | 判一个直角多边形（盖子）是否可以通过平移完全覆盖另一个直角多边形（洞）。 | 盖子覆盖住洞后，通过微调一定会有两条边重合。枚举他们各自一条水平的线段假设他们重合，再维护盖子相对于洞水平移动哪些距离是可行的，这可以用一些区间的集合来表示。然后从上到下扫下来，进行一些区间的操作即可。 | 时间：  空间： |
| 2003F | Combining Images | 一张的01图用这样的形式表示（串s是答案）：  如果当前区域全是一种颜色，s+=’1’+当前色  否则s+=’0’，递归4个象限。  给你2幅图，s1与s2,求他们的并s3  长度l<=100 | 递归着做，模拟一下。 | 时间：  空间： |
| 2003H | Combining Images | N个车站连成1排，有k个地铁出发，你可以在一个站乘上一辆地铁然后到某个车站下。求在m时刻到达n号车站的最小等待时间（在车站上等车）。  N<=50,m<=200,k<=100 | Dp,f[i][j]在i分钟时，在j号车站的最小等待时间 | 时间：  空间： |
| 2004C | Image Is Everything | 木块有颜色，给出一个方木块堆的6视图，求最多有多少木块。 | 6各面都往里推，遇到不符合的就判定为空 | 时间：  空间： |
| 2004D | Insecure in Prague | 现在要把一个长为m的串加密到长为n。方法如下：选择。然后从s开始每次跳i格把串一位位放进去。然后再从t开始，每次跳j个把串一位位放进去。剩余的空格字符随机。给出加密串，求原串。  2<=n<=40 | 预处理确定时第几位对应原串第几位，用来加速。然后枚举I,j,s,t，验证一下。加些优化就能过了。 | 时间：  空间： |
| 2004E | Intersecting Dates | 给出n+m个区间，求有几个数在m个区间中出现过且不在n个区间中出现过。  N,m<=100 | 离散，用数组模拟覆盖。 | 时间：  空间： |
| 2004F | Merging Maps | 给出n幅地图，2幅地图可以合成1幅，要求它们合成时公共标志性建筑数最大。每次将拥有最大数量的1对合成。求最终地图。  N<=10, 地图长宽 l,h<=10 | 模拟。 | 时间：  空间： |
| 2004G | Navigation | 有n个移动的物体，和一个你要确认的点。给出物体离确认点的距离，距离会有0.1的误差。求确认点。  N<=10 | 可以理解成求n个圆的交。  有一个特判：  如果所有的圆都在0.1的误差内重叠，则圆半径<0.1则答案为圆心，否则答案无穷多。  找出2个不重叠的圆，找出交点（如果两圆相差不到0.1，和普通的圆交一样做就行了），然后去判一下交出来的2个点就行了。 | 时间：  空间： |
| 2004H | Tree-Lined Streets | 现在要在大街两旁种树，有2个条件：   1. 在一条大街上，每两棵树之间的距离至少为 50 米 2. 树与它所在的大街上十字路口的距离应该不少于 25 米   求最多的树。  街道数n<=100 | 对于每条街，用其他街与它交一下，贪心地往相邻的交点里放树就行了。 | 时间：  空间： |
| 2004I | Suspense | 两栋相隔d的楼，楼中有猫和鸟。你要在左楼的s到右楼的t间连一条呈自然下垂的绳子（二次函数），最低点向下1m处有块水平的板。猫可以向上跳0.5米或者向下跳3米，通过板吃鸟。在猫吃不到鸟的情况下，绳子最长为多少。  楼高 n<=25。 | 先求出绳子的最低点。二次函数的长度等于 对积分。 | 时间:  空间: |
| 2005B | Simplified GSM Network | 平面上有n个基站，m个点,mm条边。你正在使用的基站是离你所在地最近的。两个点之间的距离为变换的基站的次数。q个询问，每次问你通过边从i号点到j号点最少变换次数。  N<=50 , m<=50, mm<=250 , q<=10 | 对于一条边。枚举每个基站，求出由它统治的区域，若非空则说明变换次数+1。就这样把权值都求出来，floyd。 | 时间：空间： |
| 2005C | Traveling | 一张n点m条边的图上，有nn个人从各自的点到s点去。路上这些人可以一起拼车。求走的路总长的最小值。还要求出每个人的路线。多种方案下，找整张图走过的点最少的。若还有多种，再将走过的点排序后字典序最小的。  N<=20 , nn<=10 | 二进制枚举那些点被走了，然后做生成树。 | 时间：  空间： |
| 2005D | cNteSahruPfefrlefe | 有一个人在洗牌。每次都将52张牌对半，然后把它们合成互相交叉的形式。现在这个人每次洗牌都有可能失误，把相邻的2张牌对调了。失误不会大于1次。给出最后的牌，求最少失误次数及方案。  操作数n<=10 | 预处理一张牌在i回合内最少失误几次才会到那个位置。暴搜时用这个优化。 | 时间：  空间： |
| 2005E | Lots of Sunlight | 有n栋楼。太阳从最左边顺时针绕过，q个询问，问你某栋楼的某层被太阳完全照到的时间。  N<=100 ， q<=1000 | 每次所有楼都看一遍，找个区间交。 | 时间：  空间： |
| 2005F | Crossing Streets | 一张图中有n条水平于坐标轴的线段，要你求从一个点到另一个点的最少穿越线段次数。你不能沿着线段走，或者穿过两条线段的交。若两条线段重叠，只被认为是穿过了1次。  N<=500 | 离散坐标，bfs | 时间：  空间： |
| 2005G | Tiling the Plane | 有一个n个点的直角多边形（边都平行于坐标系），问能否只通过复制和平移操作使得它能够铺满整个平面。（不能重叠）。多边形边长为整数，且总和Len<=50,N<=50。  一个多边形当且仅当满足以下两个条件中至少一个时可以铺满平面： 　　1. 在多边形边界上顺次存在四个点A,B,C,D（不一定要是多边形的顶点），使得A到B的边界与D到C的边界重合，B到C的边界与A到D的边界重合。这表明这个多边形可以用棋盘覆盖的方式铺满平面。 　　2. 在多边形边界上顺次存在六个点A,B,C,D,E,F（不一定要是多边形的顶点），使得A到B的边界与E到D的边界重合，B到C的边界与F到E的边界重合，C到D的边界与A到F的边界重合。这表明这个多边形可以用蜂巢覆盖的方式铺满平面。 | 让4个点的情况当做6个点一起处理，方便一些。  能成为边界上的6个点的，只可能是那些拥有整数坐标的点。可以暴力枚举前4个点，然后判断是否可行即可。 | 时间：  空间： |
| 2005H | The Great Wall Game | N\*n的格子图中有n个棋子。你要把他们移成一排。斜着也可以。每次只能移到响铃格子中。求最少移动次数。  N<=15 | 横着竖着贪心，斜着状态压缩dp。 | 时间： |
| 2005I | Workshops | N个物品a，m个盒子b，物品盒子都有2个属性x和y。只有盒子的两个属性都大于等于物品的，他们才可以配对。现要求配对数最大（都只能用一次），在配对书相同时要已匹配的物品x属性之和最大。 | 做法：将物品以x为关键字降序排序，每次找未匹配的盒子中x比那个物品大的y最小的那个，然后配对。  证明：如果未配对，却硬要配对，则需要有物品放弃配对给它。那么x属性之和会降低，而剩余的盒子却一点也没变化。所以不会变的更优。  如果配对了，却硬不要配对，后果也类似。 | 时间：  空间: |
| 2005J | Zones | 有n个圆，每个圆里有一定的人数。圆之间有m个相交区域，给你包含相交区域的那些圆和区域内的人数。要你选d个圆使得包含总人数最大。  N<=20,m<=10 | 和圆一点关系都没，直接暴搜。 | 时间：  空间： |
| 2005A | Low Cost Air Travel | 有n个城市，m张机票。每张机票都有一个价格pi,都是a1->a2->a3...->ax这种形式且x不超过10。你只能在a1使用这张机票，在途中可以下机，一下机就算这张票用完了。有q个询问，每次要你顺次经过b1,b2…bx(x<=10)，这些点，飞机路过时也算经过。求最小费用与方案。  M,q<=20 | 表示已经路过前i个点了，现在在j这个城市的最小代价。再用最短路。 | 时间：  空间： |
| 2005B | Remember the A La Mode! | 有n种饼，每种ai个，m种冰淇淋，每种bi个。I号饼配j号冰激凌可获得vij的价值。求最大价值。  N,m<=50 | 费用流。 | 时间：  空间： |
| 2005D | Bipartite Numbers | 求比n大的，是n的倍数的最小的二段数。二段数是形为  aaaa….aaaaabbbb….bbbb  < x个a > <y个b>  1<=n<=99999 | 暴力枚举 总长度,x,a,b，判断是否是n的倍数。再加上对n是10,5,2的倍数的特判，用来加速即可。特别是n是10的倍数，可以知道b一定是0，且y不会很大。 | 时间：  空间： |
| 2005E | Bit Compressor | 定义一种压缩二进制串的方法：如果将最大的连续n个1替换成n的二进制表示后长度减小了，那就换掉。已经被换过的位置不能再换。给你原串的长度，1的个数，和压缩后的串，求是否可以唯一的找出原串。  原串的长度l<=16000  压缩后的串长度n<=40 | 暴力。被替换的东西原来旁边一定夹着2个0。对于两个0之间的串，可能是原串，也可能是替换后的数字。 | 时间：  空间: |
| 2005F | Building a Clock | 有n个齿轮，齿轮之间可以共轴或者相切。要求匹配出2个特定转速的方案。不能有3个以上的齿轮共轴。  N<=6 | 总可能数很少，直接搜出每种情况，判一下。 | 时间：  空间: |
| 2005G | Pilgrimage | 有一群人有一些公共财产，有n个操作，假设现在有x个人，财产为y：   1. IN k （k<=20）：有k个人加入了，这时y一定要是x的倍数，这k个人每人交出元。 2. OUT k(k<=20)：有k个人走了，这时y一定要是x的倍数，这k个人每人得到元。 3. COLLECT k(k<=200)：每人交出k元作为公共财产 4. PAY k(k<=2000)：公共财产减少k元。   求所有可能的一开始的人数。  N<=50 | 考虑一个COLLECT操作，如果y本来是x的倍数，用了这个操作还是x的倍数，所以可以无视。  考虑一个PAY操作，如果它在最早的IN/OUT之前或者最后的IN/OUT之后，他是没用的。在两个IN/OUT之间的PAY总和一定那时是人数的倍数。  搞出所有IN/OUT之间的PAY总和，求出人数即可。 | 时间：  空间：O( |
| 2005I | Degrees of Separation | 求n个点的图的最大点对距离。  N<=50 | Floyd | 时间：  空间： |
| 2005J | Routing | 一张n个点m条边的有向图，你要标记尽可能少的点使得1通过这些点能到2,2能到1。  N<=100 | F[i][j]表示从1走到i，从j走到1，的最少点数。转移是枚举x,y，假设它们公共的线路是x->y，  经过优化后变成： | 时间：  空间： |
| 2008B | Always an integer | 给出一个多项式/D的表达式，问x为整数时是否满足表达式的值恒为整数  最高次<=100 | 随机x的值，求%d的值。 | 时间：？  空间： |
| 2008C | Conveyor Belt | 平面上有n个不相交的，顺时针或者逆时针转动的轮子，你要从起始轮开始往轮子上绕皮带直到终轮。要求皮带不能相交，而且要保证其转动方向一致。且相邻轮子间的皮带腾空长度不大于maxdis。求最短长度。  N<=20，坐标为0-10000的整数 | 虽然n有20，但是加上剪枝的暴搜不会超时的。 | 时间：  空间： |
| 2008D | The Hare and the Hounds | 猎狗和兔子在进行公路拉力赛。参赛者（猎狗）必须找到组织者（兔子）选择的一条或多条路。他们在路上走的时候会遇到各种各样的十字路口。进入任何十字路口时（除了一些特殊的，之后会讲到），他们都应该使用主要的道路规则。主要的道路规则总是“尽量直”，意思是转最小的角度（可能不需要转弯）。如果两条路转的角度相同（例如 'Y'字路口），这时主要的道路规则要求选择右边那条。  在一些路口，兔子可能违反的主要道路规则（但不会去来的那条路），随机选一条路走，这些十字路口被兔子标为选择点（通常在十字路口的路面上放一个彩色标记）。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（通常是兔子甩了一些面粉在路上）。下列都表示一个不正确的路线选择：  1.猎狗走了maxdis后还没有碰到确认点。  2.猎狗在碰到确认点前遇到了死胡同（只有一条路的十字路口）  3.猎狗在碰到确认点前遇到了其他的选择点（兔子总是会在到达下一个选择点的路上放确认点，且兔子不会再回到之前那个选择点）  在遇到一个不正确的路线后，猎狗必须沿原路返回，然后选其他路，即使中间他到达了终点，他也会忽略。  当猎狗在选择点时，他使用主要道路规则的方式略有不同，第一道路的选择是与原来一样的。但是如果猎狗必须返回这个选择点时（例如遇到一个不正确的路线），猎狗会从回来的方向使用主要道路规则选路（忽略所有他已经尝试过的路与一开始来这个点时走的路），这个过程会不断的循环知道他找到了正确的路。这种不同的主要道路规则的方式只会用在选择点上，对于本问题，猎狗不会记得它走向一条路的结果，即使他在探索选择点的时候不断的经过这条路。  对于这个问题，你会得到一个路线图(一个配置的道路和路口)，选择点的列表，确认点的位置，maxdis的值，起始点与终点（不是选择点），离开起始点时的方向。有了这些信息，你就可以模拟猎狗追兔子时的路线，你可以认为猎狗使用这些策略总可以找到兔子的路线。 | 纯模拟 | 时间：  空间： |
| 2008E | Huffman Codes | 给你一棵由n个叶节点的哈弗曼树的形态，假设哈弗曼树中的一个节点的左儿子权值小于等于右儿子。权值是1-100之间的整数，总权值=100，且求这n个叶节点有多少种不同的权值方案。  N<=20 | 从根开始扩展，根的权重为100，每次扩展时选择权重最大的那个点，确定他左儿子的权重。由于哈弗曼树生成的特性，这个点分裂出的两个儿子的权重一定要不大于现在所有点的权重。 | 时间：  空间： |
| 2008F | Glenbow Museum | 一个直角多边形，他可以用一个角序列来表示，R表示90°，O表示270°。如果一个直角多边形内有一点可以看见整个多边形，则称它是好的。求长度为n的好的角序列。  N<=1000 | 可以知道R与O的个数差4。如果角序列中连续出现2个O，则它不是好的。枚举这个角序列开头有几个R，组合数算一下。 | 时间：  空间： |
| 2008G | Net Loss | 给你一个n阶多项式g(x)，在-1到1的定义域内，你要找到一个由两条一次函数p1(x)与p2(x)组成的分段函数p(x)。这两条一次函数交于横坐标为c的一个点上。一条向左一条向右。如图：    要求最小化 | 在两条一次函数交点y坐标确定的情况下，d(p1,g)= 与P1的斜率成2次函数。  d(p,g)在斜率最优的情况下，与y坐标也成2次函数。 | 时间：  空间： |
| 2008I | Password Suspects | 要求一个包含n个串的长为L的串有几种。若不大于42则全部输出。  N<=10,L<=25,每个串长度<=10 | 对他们建AC自动机，DP。求方案的话从末尾状态开始一层层枚举上一层状态和转移暴搜上去。虽然复杂度高达27亿但是很多都是空的 | 时间：  空间： |
| 2008J | The Sky is the Limit | 有n个等腰三角形，底边在一条直线上。问这些三角形的上边界组成的轮廓的总长度。 | 1. 先把被包含的三角形去掉，方便处理。对于一只三角形，把其他的三角形拿来和他交一下，搞出轮廓在这个三角形上的长度。此方法特判较多，比较麻烦。 复杂度 2. 搞出山两两之间的交点，把横坐标离散，搞出每个横坐标下最高的山的点。相邻横坐标的点如果他们都不是地面则他们之间的连线一定是山上的一段。 复杂度 | 时间：  空间： |
| 2008K | Steam Roller | 一个n\*m地网络图，一只拖拉机要从a点走到b点。如果拖拉机从起点出发或者到达终点或者转弯，则它在此路上花的时间要\*2。求最短路。  N,m<=100 | Spfa | 时间：  空间： |
| 2009A | A Careful Approach | 有n架飞机，第i架飞机可以在ai到bi的时间内降落。你要安排这n架飞机降落的时间，使得相邻降落的飞机的间隔最小值最大。  N<=8,0<=ai<=bi<=1440 | 二分答案，暴搜n架飞机的降落顺序，贪心的去验证一下。 | 时间：  空间： |
| 2009B | My Bad | 一个g个点的无环逻辑运算图（仅含or,and,xor,not），每个节点代表一个运算，现最多有一个节点出现错误，错误有3种：只输出1，只输出0，输出相反值。有几组输入和输出，将其找出。 | 枚举那个点错的，验证一边。 | 时间：  空间： |
| 2009D | Conduit Packing | 给你4个圆的半径。圆可以任意移动。要求一个半径最小的圆把他们包住。 | 二分半径，暴搜这些圆的放置顺序，将这些圆贴着大圆的边放一定是最优的策略。 | 时间：  空间： |
| 2009E | Fare and Balanced | 一个拓扑图，修改一些边的权值使得从1号点到n号点的所有路的权值都一样。要满足一条路上不能有2条边被修改。  N<=50000 边数m<=50000 | 如果1无法走到x或者x无法走到n，就忽视x这个点。预处理dis[x]为x到n的最长距离。记f[x]为x到n的所有路径是否权值都相同。  记require[x]为1-x的所有路都必须为require[x]。  然后从1开始做，假设做到x，正在更新y，如果f[y]为真，那么将x-y的边设成dis[1]-require[x]-dis[y]，y就已经合法了。否则因为y之后一定要改边权，require[y]必须设成require[x]+c[x,y]，如果一个点的require有冲突，则无解。 | 时间：  空间： |
| 2009F | Deer-Proof Fence | 有n个点，你要建最小长度的篱笆使得篱笆外的点离这些点的距离都大于r。  N<=9,r<=200 | 预处理出包住一个集合的点所需的最短篱笆长度。求一个凸包，算下周长。所有角上的圆弧加起来是360度。  然后dp一下。 | 时间：  空间： |
| 2009G | House of Cards | Axel和Birgit喜欢玩这样的一种纸牌游戏：他们建造一个由纸牌组成的房子，当他们添加纸牌到房子的时候，会获得（或失去）游戏的分数。由于他们都有一双灵巧的手，纸牌组成的房子是不会倒塌的。他们使用半副标准纸牌。一副标准的纸牌有4种花色，2种是红色，2种是黑色。Axel和Birgit只使用2种花色，1红1黑。每种花色有13个等级。我们使用记号1R，2R，…，13R，1B，2B，…，13B来表示等级和颜色。 　　开始前，玩家要选择半副标准纸牌的一个子集，子集中所有纸牌的最大等级是M。洗完选出的纸牌后，他们从牌堆的最上面拿出8张，从左到右连续地放置它们形成4个“山峰”。举个例子，如果M=13而且前10张纸牌（26张的前10张）是： 　　6B 3R 5B 2B 1B 5R 13R 7B 11R 1R … 　　那么这个游戏开始的时候就像图7所展示的那样。 http://www.tsinsen.com/RequireFile.do?fid=E9Fd7h43 　　剩下的纸牌正面朝上被放置成一排。 　　每个玩家被认定一种颜色，红色或黑色。Birgit总被认定是黑色，Axel总被认定是红色。第一张用于组成山峰和山谷的纸牌的颜色决定了哪个玩家先开始。图7的那个例子，Birgit先开始，因为第一张纸牌是6B。 　　玩家交替进行操作。一步操作包括从一排纸牌的最前面抽取一张纸牌然后进行下列的一条： 　　1．持有这张纸牌直到下次操作（这是一张“被持有的纸牌”）。 　　2．用刚抽取的纸牌或被持有的纸牌覆盖在两个山峰之间的山谷，形成一个“基底”。如果还剩下一张牌，那么这张牌就被持有。 　　3．把2张纸牌放在基底上面，形成一个山峰（其中一张纸牌一定是一张被“持有”的纸牌）。 　　不是所有的选择总是可行的。任何时候最多持有1张纸牌，所以第一个选择只有当这个玩家没有持有纸牌时才可行。 　　因为排成一排的纸牌是正面朝上，所以两个玩家在纸牌被抽取前就事先知道纸牌的顺序。 　　如果玩家通过添加了一个基底组成了一个向下的三角形，或者通过添加了一个山峰组成了一个向上的三角形，那么玩家的分数就会像下面描述的那样更新。组成三角形的3张纸牌的等级之和将被增加到那个颜色与3张纸牌的多数颜色相等的那个玩家的分数上。如果在游戏中没有组成三角形，两个玩家的分数保持不变。 　　图7的那个例子，如果Birgit放置她的纸牌（11R）到中间的山谷上，她将获得14分。如果她放置她的纸牌到左边的山谷上，Axel获得19分。如果她放置她的纸牌到右边的山谷上，Axel获得29分。 　　如果在某步操作结束后没有纸牌等待被抽取，这个游戏就结束了。如果某个玩家在这个时候持有纸牌，那个玩家的分数将会增加（或减少）这张纸牌的等级如果这张纸牌的颜色与玩家颜色相同（不同）。 　　当这个游戏结束后，分数低的玩家将要支付一定数量的瑞典克朗给另一个玩家，数量等同于两个玩家的分数差。如果是平局，就不用支付。 　　你必须写一个程序读入一副被洗过的牌堆和一个玩家的名字，然后找出这个玩家最多能赢多少（或者最少能输多少），假设另一个玩家总是采取最优策略。  M<=13 | 用α-β剪枝优化的暴搜 | 时间：  空间： |
| 2009H | The Ministers’ Major Mess | 有n个大臣m个项目，每个大臣对ki个项目进行了表决，每个表决是’y’或者’n’。你需要对这些项目决定他是’y’或者是’n’，并且每个大臣都有一半以上的表决是对的。  N<=500,m<=100,ki<=4 | 2-sat，因为ki<=4,所以每个大臣最多只能错一个。如果是某个项目必须选择什么的情况，比如说x项目必须选择’n’，则x(‘y’)向x(‘n’)连边。 | 时间：  空间： |
| 2009I | Struts and Springs | 大窗户上有n扇窗户，每扇窗户有它的位置，宽度，高度，用6根杆子或者弹簧与其他的窗户或者大窗户相连。M个操作，每次将大窗户的宽度高度重置，弹簧会因为挤压或者伸展变化。求最终每个窗户的位置，宽度，高度。  N,m<=100 | 求出每个窗户的父亲，搞出一棵树。每次操作根据父亲的位置高度宽度，求出子窗户的对应值。 | 时间：  空间： |
| 2009J | Subway Timing | 一棵树，每条边的权值可以设成x或者x-60(0<=x<60),求任意两点间权值和的绝对值的最大值的最小值。 | 因为每条边的权值可以设成x或者x-60，所以从i的子树到i的路径和最大最小值是可以控制在之间的，所以答案不会超过120。二分+dp。 | 时间：  空间: |
| 2009K | Suffix-Replacement Grammars | 你有一个串，有n种变换。每次可以选择一种变换s->t，就是说如果你的串的一个后缀等于s，那么把这个串的这个后缀替换为t。s和t的长度是相等的。求你的串最多要几次可以变成最终串。  N<=100，串长len<=20 | 将每个长度下的每个串的后缀离散出来，在每个长度下做最短路。答案可能会超过int，比如下面的例子:  a->b b->c ac->ba bc->ca acc->baa ….  那么从a…a->c…c就要花超多次变换了。 | 时间：  空间： |
| 2010B | Barcodes | 有一些字符用二进制数表示，现在给你一些数字，如果一个数字与标准数字的误差在5%以内，则判为0，否则在标准数字的2倍的误差在5%以内，则判为1，也可能是不合法的。让你求出这些字符。  长度n<=150，数字大小<=200 | 求出最小的数字，以它的1.5倍为界限，左右分别为0与1，再看看合不合法即可。 | 时间：空间： |
| 2010C | Tracking Bio-bots | 有一个n\*m的网格图，有p个平行于坐标轴的宽为1的墙。有一个机器人只能往上往右走。求有多少位置机器人不能走到(n,m)  N,m<=10^6,p<=1000 | 离散，bfs | 时间：  空间： |
| 2010D | Castles | 一棵n个点的树，你有一支军队，你要从1个点开始，不能走同向同边2次的遍历完整棵树。每到一个节点，你要花pi的人数攻下来，然后花qi的人数防守。求军队人数的最小数使得攻下且防守所有的节点。  N<=100 pi,qi<=1000 | 树形dp，贪心。 | 时间：  空间： |
| 2010F | Contour Mapping | 给你一个这样的地形图，找所有高度是h的倍数的等高线总长度。  地图高n<=100，宽m<=100 | 对每个三角形分开统计。  如果一个三角形有两点高度相同且是h的倍数，就看看这两个点相对的左右两个点是否与他们高度相同。不同则说明这条边在整块此高度的点的边界，可以累加到答案。  然后统计三角形内所有的等高线长度。用相似三角形做，要注意B=C=h的倍数时，B-C这条边不能算。A是h的倍数时，B>A且C<A时，穿过A的等高线只能算一遍。 | 时间：  空间： |
| 2010G | The Islands | N个点，无相同横坐标。你要从1走到n走到1。1号点最左，n最右。你走的时候横坐标要递增（减小或增大）。除了1每个点都走了1遍。从1-n的路上你要经过s点，从n-1的路上你要经过t点。求最短路的方案。  N<=100 | Dp | 时间：  空间： |
| 2010I | Robots on Ice | 有n\*m的格子图，你要从(1,1)开始每个格子都走1次，然后回到(1,1)，你在步时，你必须在指定的点上。求方案数。  N,m<=8 | 暴力+优化   1. 看看有没有把图割成2半 2. 在指定步数之前有没有走过 3. 现在离下一个指定点怎么走都走不到 | 时间：  空间： |
| 2010J | Sharing Chocolate | 有一块N\*M的巧克力，现在要分给nn个人,每次分只能把某一块沿线切开，且每个人分得的巧克力是完整的一块（是矩形），现在给出这nn个人要的巧克力的大小ci，求是否可以恰好分给这nn个人。  N,M≤100，nn≤15 | 可以用f[i][j][k]，i是一个二进制数，表示一个满足人的集合，f[i][j][k]表示j\*k的矩形能不能恰好分给i所代表的集合的人。假定是横着切，于是我们可以枚举i的一个真子集，然后可以唯一地算出数量等于这个真子集的需求量的子矩形，竖着也一样，然后就是递归做。由于发现i的集合的总需求量与j\*k是相等的，所以可以省略j或k，又因为f[i][j]这个状态存在的前提条件是i的需求量能整除j，所以f[i][j]的状态数很少，可以用记忆化搜索来实现。 | 时间：  空间: |
| 2010K | Paperweight | 有一个由2个共享1面的3棱锥拼合而成的纸镇。中间有个芯片。你可以随意把纸镇放在地面上，要求纸镇的重心移动0.2后仍然不动。求芯片到地面的最小距离。 | 枚举3个点当底面。然后用3维的向量运算判断。还有一种情况就是4个点当底面。 | 时间：  空间： |
| 2011A | To Add or to Multiply | 两种操作： 1. 数值加a 2. 数值乘m  你要设计一系列的操作使得其满足无论输入是p、q之间的哪个数，输出必须是一个在r，s之间的数。  要求操作数最少，相同时看字典序（1操作<2操作）。 | M=1时直接做，否则2操作次数不会很多。枚举有几次2操作，贪心的从高位往低位判断放入几个1操作即可。 | 时间：  空间： |
| 2011C | Ancient Messages | 在黑白图中识别这6个字符：  图长度n<=200,宽m<=200 | 它们的洞的数量各不相同。 | 时间：  空间： |
| 2011E | Coffee Central | 一张n\*m的图上有t个咖啡店，定义两点之间的距离为它们的曼哈顿距离。有q个询问，每次要你找出一个点使得它周围d的距离内咖啡店最多。输出最多的值与字典序最小的点。  N,m<=1000,q<=20,t<=500000 | 前缀和。 | 时间：  空间： |
| 2011F | Machine Works | 有n台机器，每台必须在di天之内买下，它每天可以生产gi的利润，价格为pi，卖掉可得ri，求到第m天时的最大获益。 | 凸包优化dp | 时间：  空间： |
| 2011H | Mining Your Own Business | 一张联通无向图中，找最少的关键点使得任意一个点被删掉后，剩余的点都可以走到一个关键点。并求方案数。 | 求割点，答案就在于1个割点相邻的联通块中选。没有割点时要任选2个点。 | 时间:  空间: |
| 2011I | Mummy Madness | 格子图中有n只木乃伊。你先移动，每次可以向8个方向走一步，或者原地不动。木乃伊会选择8个方向中离你manhattan距离最近的一格。求你最多走多少步后被抓住。 | 二分答案，题目变成了你在mid步之内能不能走到一个没有任何木乃伊能走到的格子。用线段树做。 | 时间：  空间： |
| 2011J | Pyramids | 有两类金字塔（从高层到底层的边长）：   1. 1,2,3…x 2. 1,3,5..x   2,4,6…x  你有n个木块，你想用光这些木块，搭成一些不同的金字塔，要求这些金字塔高度至少为2。  如果有多种方法，取数量最少的，如数量相同，则比最大的金字塔谁大（用的木块数），如相同则比次大，以此类推。 | 金字塔的数量不超过。  打表发现如有解，则最多只有6个金字塔。Dp的时候把方案记下来即可。 | 时间：  空间： |
| 2011K | Trash Removal | 一个n边形，你要通过旋转它使得它能通过宽度最小的洞。求最小宽度  N<=100 | 最优情况下一定是有两个点卡住一条边。枚举这两个点，直接做。 | 时间：  空间： |
| 2012 A | Asteroid Rangers | 空间中有一些点，每个点都朝着一条直线以某个速度移动。两个点之间的边的代价即为其距离。求用最小代价将他们联通的方案随着时间的过去变换了几次。保证初始时没有两种最优方案  N<=50 | 将一条边的长度超过另一条边时作为一个事件点。将所有n^4个时间点按照时间排序，每次用dfs序判断一条边能否替换另一条边，同时保证仍然是树。每次替换后重建一边树。实际上案非常小。 | 时间:  空间: |
| 2012B | Curvy Little Bottles | 一个由一条多项式函数与两条垂直于x轴的直线xlow，xhigh围成的水瓶，求以d为单位为它标刻度。  函数中x的次数<=10 | 二分+定积分 | 时间：  空间： |
| 2012C | Bus Tour | 一张图，包含h个酒店，公交车公司与景点，要求设计一条从公交车公司🡪所有酒店🡪景点🡪所有酒店🡪公交车公司的线路。且保证去景点前先到的⌊h/2⌋个酒店必须是从景点回去时的先到的⌊h/2⌋。求路径的最小长度。  H<=20 | 预处理f[s][x]与g[s][x]分别表示从公交车公司与景点出发，走过了s集合的酒店，当前在x号酒店的最小路径。枚举前⌊h/2⌋个酒店，再枚举将两个集合连接起来的中间的那条路即可。注意只有1个酒店的情况。 | 时间：  空间： |
| 2012D | Fibonacci Words | 斐波纳契字序列的位字符串f定义为：  这里的+是连接字符串的操作。  给出n与字符串p，求p在f(n)中出现了几次。 | 只要维护f(n)前100000个字符与后100000个字符，发现当f(n-2)的长度大于100000时，f(n)的后100000个字符都相同，前100000个字符分n为奇偶两种情况。 | 时间：  空间： |
| 2012E | Inﬁltration | N个点的有向图，其中任意一对(I,j)满足要么i向j连边，要么j向i连边。若你选了一个点，则它所有直接连出去的点与自己都会被控制。要你选出一些点使得所有点都被控制。  N<=75 | 如果我们每次都选择出度最大的点，再递归用同样操作做剩余没被控制的点，那么ans<=floor(log(n))，我们搜出ans<=5时的所有情况，若还没有方案，则用最先提到的方法做。 | 时间：  空间： |
| 2012F | Keys | 有一些钥匙和钥匙环。钥匙的编号从A-Z，钥匙环从a-z。钥匙环能和钥匙环与钥匙连接。钥匙只能和钥匙环连接。你可以拆除一些连接，或者连接两个物体。现在你要将a-m的钥匙与n-z的钥匙分开，并且都被连成了一个整体。求最小的钥匙与钥匙环间的操作，其次要最小钥匙环与钥匙环间的操作。  保证它们不会连成环。 | 如果一个钥匙环既包含了a-m的钥匙，又包含了n-z的钥匙，那么肯定只能保留一种。我们用二进制暴搜它取哪一种，然后进行树形dp。有一个特判：如果当前所有的同一种类型的钥匙都没摘掉了，那么它就要一些钥匙环把它们串起来。 | 时间：  空间： |
| 2012G | Minimum Cost Flow | 空间中有n个点，每个点上有几个洞，有一个点是起点，还有一个是终点。你可以在两个不同的点之间用那些洞当插头连边，边的费用为两个点的直线距离。原图中已经有一些边存在。在起点处，你可以把水压加到任意的数值使得水可以通过管子向上爬，超过水压的那些点就流不到了。凡是流到的点，都不能有洞存在（你还可以话0.5的代价堵住一个洞）  求将起点与终点联通的最小代价。  N<=400,原来边数<=50000，点均为整数坐标。无重点 | 显然多出来的洞应当用堵而非用管子（由于整数坐标），枚举高度到那个点后，点上的洞成了问题最麻烦的地方。  先将那些被原来的边联通的点称为一块。  显然代价最优的情况下，一条从起点到终点的路一定是：…A块上的起点🡪A块上的终点🡪B块上的起点🡪 B块上的终点…  且没有一个点走了多次，这样每个点只有1个洞有用。  然后就是最短路，一条不同块的a🡪b的边，费用为。要特判洞的个数=1的块。 | 时间：  空间： |
| 2012I | A Safe Bet | R\*c的网格中有n块’/’与’\’的镜子。光线从第一行最左端向右射入，要求向最末行先右射出，求加一面镜子的方案数与字典序最小的方案。  若某个格子既可以’/’，又可以’\’，算一种。  R,c<= , n<=400000 | 显然一个格子是不可能有两个方案的。  光路显然是这样的：起点🡪…🡪某个镜子A🡪新的镜子🡪某个镜子B🡪…🡪终点。  扫描线+线段树。 | 时间：  空间： |
| 2012K | Stacking Plates | 有n个圆盘组，每个圆盘组内有ai个从上到下半径不减的圆盘，每次操作可以从某处把一个圆盘组拆成两个，或者把一个圆盘组放在另一个的头上，同时满足半径不减。求将它们全合成一个的最小操作次数。  N<=50,ai<=50 | 可以先将所有圆盘全拆出来，然后算最多有多少对圆盘使得他们属于同一个圆盘组且在最终答案内它们相邻（相当于少拆+合各一次）。贪心的搞就可以了。 | 时间：  空间： |
| 2012L | Takeover Wars | 两个人各自拥有一堆数字，两个人轮流操作，每次可以把自己堆中的两个数字并成他们的和，或者可以用自己堆的数字x，灭掉对方堆的数字y（x>y），没有数字的那个人输。保证操作时不会出现平局，问先手是否必胜。 | 2个结论：如果当前你想合并，一定是把你最大的两个数字合并，合并其他的只会被更早的灭掉。如果你最大的数字比对方最大的两个数字之和都要大的话，无论现在谁操作你都能赢。  分几种情况：   1. 你最大>他最大 且 你最大>他最大+他次大 🡪 你胜 2. 你最大>他最大 且 你最大<他最大+他次大 🡪 你把他最大灭掉后两人不停的合 或者 两人不停地合 3. 你最大<他最大 🡪不停的合 | 时间：  空间： |
| 2013A | Self-Assembly | 在平面内有n种正方形的分子，分子可以任意旋转翻转，它的四条边这些种类：一个大写字母加’+’或’-‘；或者是’00’。  一个大写字母加’+’或’-‘这类边只能和与其符号相反的边匹配。  ‘00’这类边不能与任何边匹配。  这些分子可以互相组合，只要满足相邻的边都可以匹配即可。  求这些分子是否可以无限组合。  N<=40000 , | 为了能组合出无限数量的分子，肯定是条状的最优，所以只要找到一串分子，且满足它们通过复制与旋转可以自我拼合即可。  我们把边当做点，分子当做边。只要满足一种类型的边通过一些分子能回到自己就算是无限。 | 时间：  空间： |
| 2013C | Surely You Congest | 一张n点m边的图，有c个人在某些点上想去1号点。你可以安排这些人去1号点的路径，不过这些路必须是最短的路。这些人的速度都是相同的。如果两个人在同一时间在同一条路同向前进，那么就会发生拥堵。你可以让一些人永远不动，使得最多的人没有遇到拥堵就到了1号点。  N<=25000 m<=50000 c<=1000 | 拥堵只可能发生在离1号点距离相同的那批人上，将人分为几批，每一批分别作网络流增广。 | 时间：O(c\*(m+n))  空间：  O(m+n) |
| 2013D | Factors | F(x)为x所有的分解形式，如：  20=2\*2\*5  2\*5\*2  5\*2\*2  F(20)=3  求f(k)=n的最小的k  保证n,k<2^63 | 暴力 | 时间：？  空间： |
| 2013E | Harvard | 数据是由“内存库”来组织，每个“内存库”拥有相同数量的数据。每一个访问数据的指令都由2个数控制。一个数a（非0即1）。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（bank select register “内存库”选择寄存器）中选择的“内存库”。另一个数f表示访问该“内存库”的第f个变量。我们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的序列，每个操作是： 　　●一个变量访问操作，写作Vi，i是一个正整数。 　　●一个循环操作，写作 Rn <program> E，n是一个正整数，<program>是一个任意的程序。这个操作等价于依次执行n遍<program>。  你的工作是决定一个程序最小的运行时间。更确切的说，给出“内存库”的个数和大小，需要执行的程序，输出为了执行这个程序最小的指令数（包括数据访问指令和设定BSR的指令）。为了完成这个，你必须设定一个变量到“内存库”的映射，使得这个程序运行时间最短，并且输出这个时间（也就是程序运行的指令数）。开始的时候BSR的值为undefined，直到一条命令显式的设定了它的值。  程序操作数n<=1000 ，  1≤b≤13代表“内存库”的个数  1≤s≤13代表每个“内存库”的大小（即能储存的变量数）  在数据访问Vi中，1≤i≤min(b\*s,13) | 先确定哪些变量去0号内存区，再预处理出相邻变量的对数。然后用剪枝后的搜索即可。注意到搜索量不会超过第13个[贝尔数](http://oeis.org/A000110)（ B\_13=27644437 ），10s时限下这个算法是不会TLE的 | 时间： |
| 2013F | Low Power | 有n个组，每个组有2个机器，每个机器要k个电池，其价值为电池中最小的那个。给出2nk个电池，要你分配给这些机器，使得所有组中2个机器价值只差的最大值最小。 | 二分+贪心 | 时间  :  空间：  O(n) |
| 2013H | Matryoshka | 有一行n个单层套娃，大小为ai，每个套娃可以直接放入更大的单层套娃中。每次操作可以合并相邻两个套娃，合并时打开并关闭1个套娃的代价为1，套娃合并后不可拆分。问把这些单层套娃合并成若干个完整套娃（大小为从1开始的连续正整数，层数若干）需要的最少代价。 | 两个套娃合并，设两套中分别的最小值中较大的值为x，则代价为总数量-另一套中大小<x的套娃数量。这个通过预处理一段中的最小值，以及前i个套娃中大小<=j的数量后可O(1)算出。用f[i][j]表示[i,j]段套娃合并所需的最小代价，转移为f[i][j]=min{f[i][k]+f[k+1][j]+cost(i,k,j)}。最后把完整的套娃再DP一遍即可。 | 时间：  空间： |
| 2013I | Pirate Chest | 一个n\*m的水箱，(I,j)格离水面的高度为aij，现在要放一个长最多为a，宽最多为b的立方体进去。立方体的边要与坐标轴平行。东西放进去水会满出来。求能放入的最大体积的立方体。 | 枚举立方体的上下边在哪里，然后横着扫一遍求最大的宽度，再用方程解一下高度。 | 时间：  空间： |
| 2013J | Pollution Solution | 求一个贴着x轴的半圆与一个在x轴上方的n边形面积交。  N<=100 | 把多边形在半圆内的轮廓求出来，像求多边形面积那样把这个轮廓的面积求出来。 | 时间：  空间： |
| 2013K | Up a Tree | 树的前序中序后序的子程序调用的语句有错误。已知前序中序后序在总的6次调用中各出现了3次。给出一棵树在该代码下做出的树前序中序后序结果，求出代码的所有可能情况和该情况下原树的最小字典序的前序中序后序结果。  树大小n<=26 | 枚举所有的代码可能情况，状态(i,j,0..2,0..2,0..2)表示在读入的前序串的i-j子树中，读入的前序串现在在前序/中序/后序代码中，读入的中序串和后序串的位置是唯一对应的。0..2的状态与前序的类似。然后枚举左子树的大小，记忆化搜索。 | 时间： |