IOI2014 中国国家集训队作业
试题泛做

周子凯

索引

1998

1999

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

1998 A - Crystal Clear

题目大意

在网格点阵中，相邻两个格点距离为 1，以每个格点为圆心，0.5 为半径作圆。在网格中画一个 n（n<=25）个顶点的多边形，每个顶点都在格点上，坐标绝对值 P<=250。求所有“不被破坏”的圆被多边形包含在内部的面积。若一个圆被多边形的一条边穿过，且这条边不过圆心，则这个圆被破坏。

算法关键字

计算几何

算法讨论

依次计算每个圆对答案的贡献，排除掉圆心在多边形外部的圆后，判断这个圆是否被某条边穿过。若被某条不过圆心的边穿过，则这个圆对答案不作出贡献；除此之外，若不被任一条边穿过，则这个圆对答案的贡献为整圆面积；若圆心被穿过，且圆心不在多边形顶点上，则这个圆对答案的贡献为半圆面积。对于所有圆心在多边形顶点上的不被破坏的圆，用反三角函数算出该顶点的夹角后将扇形面积统计入答案中。

时空复杂度

时间：$O(P^2n)$

空间：$O(P^2+n)$

1998 B - Flight Planning

题目大意

一架飞机需要一次飞过 n（n<=100）个航段。飞机的空速是 VCRUISE。当飞机以海拔为 AOPT 的高度飞行时，每小时将消耗燃料 GPHOPT 加仑。如果飞机不在 AOPT 高度飞行，每低于或高于 AOPT 1000 英尺，每小时消耗的燃料就会额外增加 GPHEXTRA 加仑。飞机的起飞前的高度为 0。 飞机的飞行高度每提升 1000 英尺，燃料就会消耗 CLIMBCOST 加仑（飞行高度下降是不需要消耗燃料的），飞行高度变化是瞬间完成的。每一航段的风速不是确定的，是一个关于飞行高度的线性函数。给出 n 个航段的信息，求飞机的最小耗油量以及字典序最小的方案。

算法关键字

动态规划

算法讨论

用状态 f[i][j]表示飞完前 i 个航段，第 i 个航段飞行高度是 20000+20j 时的最小耗油量。转移时枚举下一航段的飞行高度，算出风速、飞行速度、耗油量后进行转移。

时空复杂度

时间：O(n)

空间：O(n)

1998 D - Page Selection by Keyword Matching

题目大意

一个网页有不超过 8 个标签，一个搜索指令有不超过 8 个关键字，标签和关键字的权重从前往后从 8 开始依次递减，一个网页与一个搜索指令的匹配程度是所有匹配的标签和关键字的权重的乘积的和。给出 n（n<=25）个网页和 m 个搜索指令，输出每个指令匹配度最高的 5 个网页。

算法关键字

模拟

算法讨论

直接模拟，对于每个搜索指令，计算每个网页与这个指令的相关度，按相关度从大到小排序输出前五个。

时空复杂度

时间：O(nmlogn)
空间：O(n+m)

1998 E - Petri Net Simulation

题目大意

有 NP（NP<100）个变迁和 NT（NT<100）个库所，每个变迁有若干输入库所和输出库所，每个库所内有若干令牌。若一个变迁的每个输入库所都有至少 1 个令牌，则这个变迁是可用的。有 NF（NF<1000）轮操作，每轮操作可以选择若干可用的变迁使其发生，发生时，它的所有输入库所内的令牌数减 1，所有输出库所内的令牌数加 1。求 NF 轮操作后每个库所内有几个令牌，或是判断无法进行完 NF 轮操作。保证无论每轮操作选择哪些变迁，最终结果都是一样的。

算法关键字

模拟

算法讨论

因为题目保证了最终结果的唯一性，所以每次操作选第一个可用的变迁进行模拟，若没有可用变迁则判断为无法完成。

时空复杂度

时间：O(NP*NT*NF)
空间：O(NP*NT)

1998 G - Spatial Structures

题目大意

一张黑白图片可以用 01 矩阵表示也可以用一棵四分树表示，写一个程序实现图片表示的两种形式之间的转换。

四分树中，若一个节点对应区域是全黑或全白的，则这个节点标记上黑色或白色，且不再分割，否则将这个区域分成四个等大的子区域，重复此过程。一棵四分树是用根节点到所有黑色叶子节点的路径的序列来表示的。

图片的边长 n<=64，且一定是 2 的幂。

算法关键字

模拟 数据结构

算法讨论

对于 01 矩阵转四分树的情况，先预处理 sum[i][j]表示(1,1)到(i,j)组成的矩阵的数字和。处理一个区域时，若发现这个区域内的数字和等于区域内的元素个数，则说明这是一个全黑的区域，无需继续分割；若区域内数字和等于 0，说明这是一个全白的区域，也无需继续分割；否则递归处理四个子区域，最后将所有黑色子节点的路径用五进制表示后排序输出。

对于四分树转 01 矩阵的情况，我们把给出的十进制数转成五进制数，就可以知道这条路径对应的是哪个区域，将它全部染黑，最后就能得到整个 01 矩阵。

时空复杂度

时间：$O(n^2 \log n)$

空间：$O(n^2)$

1999 A - Bee Breeding

题目大意

蜂巢状的六边形网格中，中心被标记为 1，从中心开始螺旋状逆时针将其与格子标记为 2,3,4,5……相邻的格子距离为 1。多组询问，每次询问标号为 a 的格子与标号为 b 的格子的距离。（a,b<=10000）

算法关键字

分类讨论+简单计算

算法讨论

设最大标号为 n，询问组数为 q。先在 O(n)时间内求出每个标号所在的格子的坐标，回答询问时讨论两个格子的相对位置关系，可以在 O(1)时间内得出答案。

时空复杂度

时间：O(n+q)
空间：O(n)

1999 C - A Dicey Problem

题目大意

n*m（n,m<=10）的网格内，每个格子为一个 0 到 6 的数字或五角星。一个标有 1 到 6 的骰子在网格上滚动，每次可以向上下左右滚一格，要求滚到的格子内的数字与当前骰子顶面的数字一样或格子内为五角星。给定起点和骰子的初始方向，求一条路径使骰子从起点出发又回到起点，若判断路径不存在。

算法关键字

宽度优先搜索

算法讨论

先打一张 6*6 的表 g，g[i][j]表示骰子前面为 i，顶面为 j 时右面是多少。之后 BFS，状态 f[x][y][i][j]表示当前处在网格(x,y)，前面为 i，顶面为 j 是否能达到，转移时记录下从哪个状态转移来就可以输出方案了。

时空复杂度

时间：O(nm)

空间：O(nm)

1999 D - The Fortified Forest

题目大意

在二维平面内有 n（n<=15）棵树，每棵树视为一个点，第 i 棵树坐标为(x[i],y[i])，价值为 v[i]，能提供长度为 l[i]的木料。现在要砍掉若干棵树，用获得的木料做成栅栏将剩下的树围起来，输出砍掉的树价值和最小的方案。

算法关键字

枚举 计算几何

算法讨论

枚举砍掉哪些树，求出剩下的树构成的凸包的周长，若获得的木料长度不小于凸包周长则更新答案。需要特判只剩下 1 棵或 2 棵树的情况。

时空复杂度

时间：$O(2^n n\log n)$
空间：$O(n)$

1999 E - Trade on Verweggistan

题目大意

有 w（w<=50）堆商品，每堆最多有 b（b<=20）个商品，每个商品有一个价值（可正可负），从每堆的堆顶选若干商品，求最大总价值，以及最优方案种可能选几个商品。

算法关键字

动态规划

算法讨论

用 f[i][j]表示从前 i 堆中选了 j 个商品获得的最大价值，每次枚举从当前堆堆顶取几个商品后转移。

时空复杂度

时间：$O(w^2b^2)$
空间：$O(w^2b)$

1999 H - Flooded!

题目大意

一个地区被描述成 n*m（n,m<=30）的网格，每一格都有其海拔高度，面积都为 $100m^2$。在这个地区内产生了体积为 V 的降水，积水能够无视地形流动到当前水平面最低的区域，求有多少格子被淹没，及最后的水平面高度。

算法关键字

模拟

算法讨论

从低到高依次处理每一个海拔高度，若发现剩余的积水无法将水平面上升到下一个海拔高度，则可以得出最终答案，否则上升水平面后继续计算。

时空复杂度

时间：O(nmlog(nm))
空间：O(nm)

2000 A - Abbott's Revenge

题目大意

有一个 n*m（n,m<=9）的网状迷宫，迷宫中每个路口的四个方向上都标有一些箭头，表示当你面向四个方向站在这个路口时，下一步分别能向哪些方向走。给出起点位置、初始朝向和终点位置，求起点到终点的最短路径，或判断路径不存在。保证答案唯一。

算法关键字

宽度优先搜索

算法讨论

先预处理数组 map[x][y][t][k]（0<=t,k<=3），表示站在点(x,y)，朝向为 t，下一步可以往哪几个方向走。之后进行 BFS，d[x][y][t]（0<=t<=3）表示走到点(x,y)，朝向为 t 的最少步数。记录每个状态的前驱即可输出方案。

时空复杂度

时间：O(nm)
空间：O(nm)

2000 B - According to Bartjens

题目大意

给出 n（n<=9）个数字，在这 n 个数字之间插入一些'+'，'-'，'*'，组成一个算式，使得答案为 2000，按字典序从小到大输出所有算式。数字不能有前导零，'-'只作减号不作负号，至少要插入一个符号。

算法关键字

搜索

算法讨论

搜索相邻两个数字减插入什么负号，或是不插入符号，除去存在前导零的不合法情况，计算结果若等于 2000 则输出。按插入'*'， 插入'+' ，插入'-'，不插入的顺序搜索能保证先搜出的结果比后搜出的结果字典序小。

时空复杂度

时间：$O(4^n)$
空间：O(n)

2000 C - Cutting Chains

题目大意

一个链环中有 n（n<=15）个环，其中某些环是套在一起的。求至少打开几个环才能将原链环变成一条链。

算法关键字

搜索

算法讨论

搜索打开哪些环，则原链环被分成了若干可自由拆卸的环和一些较小的链环。判断每个小链环是否都为链状后，若打开的环数>=小链环数-1，则说明可以拼成一条链，更新答案。

时空复杂度

时间：$O(n^2 2^n)$
空间：$O(n^2)$

2000 E - Internet Bandwidth

题目大意

有 n（n<=100）台机器和 m（m<=n*(n-1)/2）个连接，每个连接使某两台机器能以每单位时间 c 单位信息的速度通信。求机器 S 和机器 T 最大能以每单位时间几单位信息的速度通信。

算法关键字

网络流

算法讨论

本题很明显是一个最大流模型，可以直接用 sap 算法解决。

时空复杂度

时间：$O(n^2m)$

空间：$O(n^2)$

2000 F - Page Hopping

题目大意

有 n（n<=100）个网页，每个网页都可以直接链接到其它某些网页，给出所有的链接，求任意两个网页的平均链接次数。

算法关键字

最短路

算法讨论

将网页看做节点，链接看做有向边，用 floyd 算法求出任意两点间最短路后统计平均数。

时空复杂度

时间：$O(n^3)$
空间：$O(n^3)$

2001 A - Airport Configuration

题目大意

机场布局为 1*(n-1)（n<25）的网格状，网格边长为 1，乘客可以在网格线上从入口移动到出口。北面是 n 个出口，南面是 n 个入口，每个入口和出口对应一个城市。给出若干对城市之间的客流量，再给出若干个入口/出口与城市的对应方案，求出所有方案中每对入口出口间客流量与距离的乘积的和，从小到大输出。

算法关键字

简单计算

算法讨论

对于给出的每种方案，直接计算出每对出入口间的距离与客流量的乘积，最后将所有方案排序后输出。

时空复杂度

时间：$O(n^2)$
空间：$O(n^2)$

2001 B - Say Cheese

题目大意

一块奶酪中有 n（n<=100）个空洞，每个空洞是一个球体。一只虫子需要从一点移动到另一点，在奶酪中移动 1mm 距离需要 10s，在空洞中移动不需要时间，求最短时间。

算法关键字

计算几何 最短路

算法讨论

本题可以转化成最短路问题。将起点和终点视为半径为 0 的球体，i 点和 j 点之间连一条权值为（i 和 j 之间的直线距离-Ri-Rj）的无向边，其中 Ri 是第 i 个空洞的半径，若边权值为负则视为 0。求起点到终点的最短路乘以 10 即为答案。

时空复杂度

时间：$O(n^2)$

空间：$O(n)$

2001 F - A Major Problem

题目大意

一个大调音阶由 8 个音符组成，每个音符是一个 A~G 的字母加上一个“#”号或“b”号（可能不加），一个音符可能有两种表示方法。一个大调音阶中， A 到 G 的每个字母将在音阶中出现恰好一次，同时第一个字母将例外地在音阶最后重复出现一次，且音阶当中不允许同时出现升调或降调记号。给出若干对大调音阶 X 和 Y，要求把 X 的某些位置的音符换成 Y 的对应位置的音符后输出，或判断不存在 X/Y 大调，或判断要求替换的音符不在这个大调中。

算法关键字

枚举 模拟

算法讨论

枚举音阶中使用升号还是降号，每次按跨度选择这个音符字母较小（规定 G 比 A 小）的表示法，就能求出给定的 X 大调包含哪 8 个音符，或判断不存在 X 大调。之后对于能在 X 大调中找到的音符直接替换即可。

时空复杂度

时间：O(1)
空间：O(1)

2001 H - Professor Monotonic's Network

题目大意

在一个比较网络中有 n（n<=12）个数和 k（k<=150）个比较器，第 i 个比较器会将第 x[i]个数和第 y[i]个数进行比较，将较小的数放在 x[i]号位，较大的数放在 y[i]号位。一个比较器完成 工作要花 1 单位时间，多个不共用数字的比较器可以同时工作，编号小的比较器总是优先工作。求整个比较网络完成比较所花的时间，并判断该比较网络是否总能把 n 个数从小到大排序。

算法关键字

随机化 模拟

算法讨论

求比较网络的工作时间可以用递推解决，f[i]表示最后一次使用数字 i 时的时刻，处理第 j 个比较器时将 f[x[j]]和 f[y[j]]均赋为 max(f[x[j]],f[y[j]])+1。对于判断是否总能完成排序，可以随机 T 次初始 n 个数的值，模拟后判断是否完成了排序。

时空复杂度

时间：O(Tn)
空间：O(n)

2002 A - Ballons in a Box

题目大意

一个边与坐标轴平行的长方体内有 n（n<=6）个点，每次选一个在长方体内，且不在其它球体内的点，以它为球心扩展出一个球体，直到这个球体与其它某个球体相切或与长方体的面相切。求长方体内最小剩余体积。

算法关键字

枚举 计算几何

算法讨论

枚举扩展的顺序后，依次计算每个球体的半径应该为多少。判断求新是否被其它球体包含后，求出当前球体与之前已扩展的球体和长方体六个面分别相切时的半径，取最小值即为实际能扩展出的半径。

时空复杂度

时间：O(n*n!)
空间：$O(n^2)$

2002 C - Crossing the Desert

题目大意

在一片沙漠中有 n（n<=20）个绿洲，可以视为二位平面中的点。1 号绿洲是起点，n 号绿洲是终点。你有一个背包，可以装水和食物，它们的总量不能超过 tot 个单位。你每行走 1 单位距离，就要消耗 1 单位水和 1 单位食物。水可以在各个绿洲处补充，食物只能在起点处补充，但你可以将背包中的食物储存在其它绿洲处。求从起点到终点最少消耗多少单位的食物。

算法关键字

动态规划

算法讨论

从终点开始逆推，设 d[i]表示从 i 点出发，最少需要储存多少食物在 i 点才能到达终点。枚举 i 的前一个点 j，为了将食物从 j 运到 i，一定是先经过若干次往返，每次消耗一共 dis[i][j]*3 单位的食物和水，背包剩下的空间用于运食物，最后将 j 点的食物全部搬空，消耗共 dis[i][j]*2 单位的水和食物。由此可以计算 j 走到 i 需要准备多少额外的食物，用 SPFA 进行转移。

时空复杂度

时间：$O(n^2)$

空间：$O(n)$

2002 E - Island Hopping

题目大意

有 n（n<=50）个岛屿，每个岛屿视为二维平面内的一个点。给出每个岛屿的坐标和岛上人口数，要在 n 个岛之间拉 n-1 条电缆，使所有岛屿都与 1 号岛屿建立连接，且要求电缆总长度最小。所有电缆同时开始建设，每单位长度需要消耗 1 单位时间，求所有居民与 1 号岛屿建立连接的平均时间。

算法关键字

最小生成树

算法讨论

求出最小生成树后，依次计算每个岛屿在哪一时刻与 1 号岛屿建立连接。将最小生成树上的边按长度从小到大依次加入图中，若加完某条边 E 时岛屿 i 与岛屿 1 属于同一联通块，说明 i 号岛上的居民在 E 完成的时刻建立连接。

时空复杂度

时间：$O(n^3)$
空间：$O(n^2)$

2002 G - Partitions

题目大意

一个矩形的划分是指把一个矩形分成若干个较小的、不重叠的子矩形。给出两种矩形划分方案，求一种边数最少的矩形划分方案，使得原来两种方案的每条边都在此方案中出现过；求一种边数最多的矩形划分方案，使得方案中每条边都在原来两种方案的其中一种中出现过。矩形边长 n<=20。

算法关键字

Floodfill

算法讨论

第一问答案显然为给出的两种划分方案取并；对于第二问，先求出两种划分方案取交的结果，但这不一定是一种矩形划分方案，所以每次用 Floodfill 找出形状不是矩形的联通块，将不合法的边删去，直到每个联通块都是矩形的。

时空复杂度

时间：$O(n^4)$
空间：$O(n^2)$

2002 H - Silly Sort

题目大意

给出 n（n<=1000）个数，每次可以交换任意两个数，代价为这两个数的和，求将所有数升序排列的最小代价。

算法关键字

简单计算

算法讨论

将数列排序后求出所有置换，对于每个置换，设其元素个数为 L，完成这个置换有两种方法，一是用置换中最小的元素依次与其他元素交换，共进行 L-1 次交换，二是引入整个数列中最小的元素，以置换中最小的元素为切入点依次进行交换，最后再将新引进的元素换出，共进行 L+1 次交换，选择代价较小的方案即可。

时空复杂度

时间：O(nlogn)
空间：O(n)

2003 B - Light Bulbs

题目大意

n（n<=332）个灯泡一字排开，标号为 1 到 n，有 n 个开关，第 i 个开关可以控制第 i-1，i 和 i+1 个灯泡（如果灯泡存在的话）。给出灯泡的不是状态和目标状态，求最少操作几个开关可以达到目标状态，或判断无解。输出操作次数最少的方案中字典序最小的方案。

算法关键字

枚举 递推

算法讨论

枚举第 1 个开关是否要操作，若第一个开关是否操作是确定的，则之后所有开关的操作也都是确定的。若操作完后最后一个灯泡的状态也与目标状态一样，则说明这是一种可行的方案。可行的方案最多两种，比较后输出较优的那一种。因为始末状态是以十进制形式输入的，需要 $O(n^2)$的时间转换成二进制。

时空复杂度

时间：$O(n^2)$
空间：$O(n)$

2003 D - Eurodiffusion

题目大意

在二维平面内有 n（n<=20）个国家，每个国家都表示为一个矩形，第 i 个国家左下角为(x1[i],y1[i])，右上角为(x2[i],y2[i])（1<=坐标范围<=10），矩形内每个整点都代表一个城市。每个国家都有自己的硬币。初始每个城市都有本国的 1000000 个硬币，每天，每个城市都会向上下左右 4 个城市一定量流通一定量的硬币，对于每一种拥有的硬币，每拥有 1000 个就流通 1 个。求每个国家分别在第几天时，国内所有城市都拥有所有国家的硬币。

算法关键字

模拟

算法讨论

直接模拟硬币的流通，直到所有国家的所有城市都拥有所有种类的硬币时模拟结束。

时空复杂度

时间：$O(Ans*n^2)$
空间：$O(n^2)$

2003 F - Combining Images

题目大意

一张黑白图像可以用四分树存储，树中每个节点表示一个子矩形，若是全黑或全白则直接标记这个节点的颜色，否则这个节点分出四个儿子节点分别表示四个子区域。一棵四分树可以用 01 串编码，若当前节点为全黑或全白，则编码首位为 1，第二位为 0 或 1 表示颜色。若当前节点有子节点，则编码首位为 0，后接四个子树的编码。给出两幅图像的四分树的编码，求出这两幅图的交的四分树的编码。编码用 16 进制表示，长度 L<=100。

算法关键字

数据结构

算法讨论

因为编码最长为 100，可知图像边长最大可达 2^57，无法直接模拟。可以先建出第一幅图像的四分树，求出第二幅图像哪些区域是白色后，用类似线段树区间操作的方法将这些区域从第一棵树中“减去”，最后求出树的编码后输出。

时空复杂度

时间：$O(L^2)$
空间：$O(L)$

2003 G - A Linking Loader

题目大意

在一个模块中有 n（n<=100）个变量定义和若干变量引用，之后有 m 个待写入内存的值，它们可能是直接给出的数字，或是以“第 k 个被引用的变量”的形式表示。引用的变量可能被定义在其它模块中。你需要写入内存的所有值的某种哈希值，以及按字典序列出所有被定义或引用的变量和它们的值，并标记被重复定义的变量。被重复定义的变量以第一次定义的值为准，引用为定义的变量其值视为 0。

算法关键字

模拟

算法讨论

读入所有变量定义和引用后处理出变量-值的对应表，之后处理所有写入内存操作，若是引用变量则去变量-值表中查询，最后计算哈希值。

时空复杂度

时间：O(nm)

空间：O(n+m)

2003 H - A Spy in the Metro

题目大意

在一条双向运行的地铁线内有 n（n<=50）个站台，有 m_1 班列车从 1 号站台出发前往 n 号站台和 m_2 班列车从 n 号站台出发前往 1 号站台。给出相邻两个站台的行驶时间和每班列车的发车时间，问从 1 号站台出发，T（T<=200）时刻准时到达 n 号站台，最少需要在站台上等待多少时间。

算法关键字

动态规划

算法讨论

用 f[t][i]表示在 t 时刻处于 i 号站台，之前最少等待了多长时间。转移时要么选择等待到时刻 t+1，要么选择乘坐刚好在此刻到站的列车到下一站。答案为 f[T][n]。

时空复杂度

时间：$O(nT(m_1+m_2))$

空间：$O(nT+m_1+m_2)$

2003 I - The Solar System

题目大意

两颗行星围绕同一颗恒星旋转，它们的轨道是椭圆形的，长轴和短轴都在坐标轴上，恒星的位置处在椭圆 x 坐标非负的焦点处。给出第一颗行星的轨道长轴长度、短轴长度、公转周期，第二颗行星的轨道长轴长度、短轴长度，求第二颗行星从 x 坐标最大处逆时针旋转 t 时间后的坐标。所有输入数据范围在 int 以内。

算法关键字

二分 计算几何

算法讨论

根据开普勒第三定律$\left(\frac{t1}{t2}\right)^2 = \left(\frac{a1}{a2}\right)^3$，可以算出第二颗行星的公转周期。因为给定的 t 有可能超过一个周期，所以先算出等价的小于 1 个周期的非负时间 T，结合整个椭圆的面积，根据开普勒第二定律算出 T 时间内行星与恒星连线扫过的面积。可以通过二分 T 时间时行星转过的弧度来算出最终坐标。因为涉及的数值可能很大，需要特别注意精度问题。

时空复杂度

时间：O($\log_2 \frac{2\pi}{eps}$)

空间：O(1)

2003 J - Toll

题目大意

你需要将一批货物从一个地点运到另一个地点。共有 n（n<=52）个地点，某些地点之间由一条双向边。地点分城镇和村庄，进入村庄需要交纳一件货物，进入城镇需要交纳当前所带的 5%（取上整）的货物。问从地点 x 运送 p 件货物到地点 y，最少需要准备几件货物。

算法关键字

最短路

算法讨论

用 d[i]表示从 i 点开始，至少要带多少货物才能完成运送。从终点开始倒推，在已知 d[i]的值和点 i 是城镇还是村庄的情况下，不难算出在进入 i 点前需要带多少货物，用类似最短路的方法转移。

时空复杂度

时间：$O(n^2)$
空间：$O(n^2)$

2004 E - Intersecting Dates

题目大意

给出两组日期的区间（每组区间数 n<=100），日期以 YYYYMMDD 的形式给出，求出第二组区间的并减去第一组区间的并的结果。

算法关键字

简单计算

算法讨论

将两组区间分别合并，使得每组内的区间互不相交，再按左端点排序后扫描一遍即可得出结果。

时空复杂度

时间：O(nlogn)
空间：O(n)

2004 G - Navigation

题目大意

给定二维平面内 n（n<=10）个移动的信号源的移动方向、速度、发出信号的时间、信号传播速度，接收点同时接收到所有信号源发出的信号的时刻 T，求接收点的坐标，输出另一个给定的点在接收点的哪个方向。或判断为无解或多解。

算法关键字

计算几何

算法讨论

在某一时刻，一个信号源发出的信号位于一个圆周上，题目的本质即是判断 n 个圆是否交于同一个点。先求出第一个和第二个圆的两个交点（无交点则为无解），再依次判断每个交点是否都在其它圆的圆周上。若一个交点同在剩下 n-2 个圆的圆周上，这个点即为一个可能的答案。

时空复杂度

时间：O(n)
空间：O(n)

2004 H - Tree-Lined Streets

题目大意

有 n（n<=100）条马路，每条马路都可视为一条线段，线段和线段的交点视为十字路口。你要沿路种一些树，保证同一条马路上相邻两棵树距离至少为 50，且每棵数距离它所在的马路上的十字路口距离至少为 25。问最多能种几棵树。

算法关键字

计算几何

算法讨论

依次处理每条马路，算出其他马路与它的交点后，将这些交点排序，依次计算相邻两个交点间能种几棵树。需要特殊处理马路两端和马路上没有交点的情况。

时空复杂度

时间：$O(n^2 \log n)$
空间：$O(n)$

2005 C - The Traveling Judges Problem

题目大意

在一张有 nc（nc<=20）个点和 nr 条边的图中有 nj（nj<=10）个特殊点，选一些边使得 nj 个特殊点通过被选的边都与点 dc 连通，求选择的边的最小权值和，并输出每个特殊点到点 dc 的路径。要求经过的不同的点数尽量少，点数相同时所有经过的点的编号从小到大排序后字典序尽量小。

算法关键字

状态压缩动态规划

算法讨论

不难发现，所选的边一定构成一棵树，树根一定为 dc，所有特殊点到 dc 的路径一定是从树上深度大的点走向深度小的点。于是可以设计这样的动态规划，f[i][sta]表示当前子树的根为 i，子树中包含的特殊点为 sta（sta 为 nj 位二进制数），所选的边权和最小为多少。有两种转移，一是将当前子树的根连向一个新的节点，二是将两棵根相同的子树合并。转移时记录此时最优的经过的点的集合，和前导状态（可能有两个）。输出方案时从 $f[dc][2^{nj}-1]$逆推一遍就可以知道每个特殊点到 dc 的路径。

时空复杂度

时间：$O(2^{2n}+nc^2*2^{nj}+nc*3^{nj})$
空间：$O(nc*2^{nj})$

2005 E - Lots of Sunlight

题目大意

一条东西方向的直线上有 n（n<=100）幢楼房，每幢楼房有各自的层数。给出每幢楼房的宽度、楼房之间的间距、每层楼的高度、日出和日落的时间，有 q（q<=1000）个询问，每次问某幢楼房的某层能完全被阳光照射的时段。

算法关键字

计算几何

算法讨论

对于一幢楼房的一个楼层，只要求出左边的楼房房顶的右端到该层层底的左端的连线与地面的最大夹角，就可以用反三角函数知道最早被完全照射的时间点，类似地可以求出最晚时间点。

时空复杂度

时间：O(nq)
空间：O(n)

2005 F - Crossing Streets

题目大意

在一个城市中有 n（n<=500）条马路，每条马路都可看做二维平面内的一条线段，保证马路与 x 轴或 y 轴平行。你需要从一个点走到另一个点（保证这两个点都不在马路上），问最少要过几次马路。

算法关键字

离散化 宽度优先搜索

算法讨论

将坐标离散化后，用 01BFS 即可求出至少要跨越几条线到达终点，即：当一个点入队时，将与这个点通过边权为 0 的边相连的点也一起入队。

时空复杂度

时间：$O(n^2)$
空间：$O(n^2)$

2005 G - Tiling the Plane

题目大意

给出一个有 n（n<=50）个顶点的直角多边形，问这个多边形能否密铺平面。多边形边长为整数，周长 C<=50。

算法关键字

枚举

算法讨论

题中给出了提示，若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，则表明这个多边形可以用蜂巢覆盖的方式铺满平面。因为 A 到 D 的边界长度必定是一半的周长，所以可以枚举 A,B,C 点的位置，算出 D 点的位置后，依次描出边界 D-E，E-F 和 F-A，看是否与原图吻合。另一种“棋盘覆盖”的密铺方式其实是包含在蜂巢覆盖中的，可以不用考虑。

时空复杂度

时间：$O(L^4)$
空间：$O(L^2)$

2005 H - The Great Wall Game

题目大意

在一个 n*n（n<=15）的棋盘上有 n 个棋子，每一步可以将一个棋子移动到上下左右的其中一个空位上。求最少几步将所有棋子移成一行、一列或对角线。

算法关键字

枚举 费用流

算法讨论

假设最终位置已确定，因为总存在一种移动顺序使得棋子在不重叠的情况下移到目标位置，所以不用考虑重叠的问题。枚举棋子的最终位置，用最大费用最大流求出最小移动步数后取最小值输出。

时空复杂度

时间：$O(n^4)$
空间：$O(n^2)$

2005 I - Workshops

题目大意

有 n（n<=1000）场会议和 m（m<=1000）个房间，每场会议有其参与人数和持续时间，每个房间有人数限制和时间限制，每个房间只能进行一场会议，求最少有几场会议无法被安排，以及在无法被安排的会议数量最少的情况下，最少有几个人无法参加会议。

算法关键字

贪心

算法讨论

可以采取如下贪心策略：按时间限制从小到大处理房间，在未安排的会议中选能被该房间接受的持续时间最长的会议，安排进该房间。

时空复杂度

时间：O(n+mlogm)
空间：O(n+m)

2005 J - Zones

题目大意

有 n（n<=20）座服务塔，n+m（m<=10）个区域，每座服务塔可以覆盖若干指定区域，每个区域有各自的收益。要求选恰好 k 座塔，使得被覆盖的区域收益和最大。

算法关键字

搜索

算法讨论

鉴于数据范围较小，可以用搜索的算法解决此题。预处理每座服务塔控制哪几个区域，搜索选择哪几座塔，取最优方案即可。

时空复杂度

时间：$O(2^n m)$

空间：O(n+m)

2006 D - Bipartite Numbers

题目大意

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 不是 0，并且 s 的所有出现均排列在所有的 t 的前面。给出密钥 n（n<=99999），n 对应的公钥是大于 n 且能被 n 整除的最小二段数。求这个公钥。

算法关键字

枚举

算法讨论

设 f(x)是连续 x 个 1 组成的数对 n 取余的值，则一个由 p 个 s 和 q 个 t 组成的二段数与 $s \cdot f(p+q)+(t-s) \cdot f(q)$关于 n 同余，本题中要求的二段数对 n 取余为 0。于是枚举 p+q、s 和 t 后，通过预处理可以很快地判断 q 是否存在，并求出最小的 q 和最大的 q 满足上式对 n 取余为 0。控制枚举顺序后第一个枚举到的解即为答案。因为公钥必须大于密钥，所以要特判密钥 n 本身为二段数的情况。

时空复杂度

时间：O(n)
空间：O(n)

2006 F - Building a Clock

题目大意

给你 n（n<=6）个齿轮和无限根转轴，每个齿轮有各自的齿数，每根转轴上最多安装 3 个齿轮。给出起始转轴的转速 R，求是否能利用这些齿轮和转轴构造出一根转速为 24 的转轴和一根转速为 2 的转轴，并输出方案。

算法关键字

搜索

算法讨论

因为 n 的范围较小，可以搜索所有的连接方案，计算出所有转轴的转速后，若存在一根转轴转速为 24 且有另一根转速为 2，则更新答案。

时空复杂度

时间：$O(n^{2n})$
空间：$O(n)$

2006 G - Pilgrimage

题目大意

Jack 是一支朝圣队伍的记账员，他的账本记录中有如下 4 种信息：有 k 人加入队伍；有 k 人离开队伍；每个人向经费中上交 k 元；从经费中扣除 k 元。若当前队伍中有 x 人，共有经费 s，则当有人加入队伍时，新加入的人每人向经费中上交 s/x 元；当有人退出队伍时，退出的人每人从经费中拿走 s/x 元。Jack 发现，每次上交或拿走的金额都是整数。给出账本中的一段记录，求在记录开头的时刻队伍中有多少人。若答案数量有限，输出所有可能的答案，否则输出可能的最小答案。

共 T（T<=30000）组数据，每组数据中有 n（n<=50）条账目，每次操作涉及的金额 k<=2000.

算法关键字

数论

算法讨论

不难发现，上交经费的操作不会影响答案，可以直接无视。因为原有经费是不确定的，所以只需要关心支出的经费是否能被当前的人数整除。因为总能有一个初始经费值能满足第一次人数变动时经费数能被人数整除，所以从第二个人数发生变动的条目开始，维护当前可能的答案有哪些。可能的答案的数量是$\sqrt{nk}$级别的，所以第一次需要维护答案时，将当前经费数的所有约数都列入可能的答案中，之后维护答案时从中删去不合法的答案。

时空复杂度

时间：$O(Tn\sqrt{nk})$

空间：$O(nk)$

2006 I - Degrees of Separation

题目大意

给出一张 n（n<=50）的无向图，求最短路长度最大的两点间的最短路长度。

算法关键字

最短路

算法讨论

用 floyd 算法求出人意两点间的最短路后，取最短路长度最长的输出。

时空复杂度

时间：$O(n^3)$
空间：$O(n^2)$

2006 J - Routing

题目大意

在一个网络中有 n（n<=100）台计算机和 m 个链接。若计算机 A 和 B 都安装了所需的软件， 且存在一条 A 指向 B 的链接，则 A 可以向 B 发送信息。两台计算机也可以通过其它中间计算机发送信息，所有涉及的计算机都需要安装软件。求至少给几台电脑安装软件，可以支持计算机 1 向计算机 2 发送信息，且计算机 2 也可以向计算机发送信息。

算法关键字

动态规划

算法讨论

将网络看成一张有向图。设有两个棋子，都从 1 号点出发，一个一直沿正向边走，一个一直沿反向边走，用状态 f[i][j]表示两个棋子分别在 i 和 j 时最少涉及到的点数。有两种转移：一是选择 i 或 j 走一步，如果 i 和 j 走到了一起则总点数不变，否则总点数加 1；二是将 i 和 j 的位置交换，点数加上 i 到 j 的最短路长度-1。需要用 SPFA 转移状态。

时空复杂度

时间：$O(n^3)$
空间：$O(n^2)$

2007 A - Consanguine Calculations

题目大意

给出父母和孩子三人中两人的 ABO 血型系统的血型（A,B,O 或 AB）和 Rh 血型系统的血型（+或-），判断另外一人可能的血型有哪些。

算法关键字

枚举

算法讨论

枚举后判断是否合法。对于 Rh 血型系统，可以直接判断，当且仅当孩子为“+”且父母均为“-”时是不合法的，其余情况均是合法的。对于 ABO 血型系统，打两张表 a 和 b，a[i][j]表示表现型为 i 的人基因型有哪些可能，b[i][j][k]表示基因型为 i 和基因型为 j 的人的孩子基因型有哪些可能，就可以判断合法性。

时空复杂度

时间：O(1)
空间：O(1)

2007 G - Network

题目大意

在网络上发送 n（n<=5）条信息，每条信息都被分割成了若干了数据包，共有 m（m<=1000）个数据包。数据包到达接收端的顺序不同于原先的顺序，所以需要暂存库来调整接收到的数据包的顺序。一个数据包可以选择被直接接收端接收，也可以选择存入暂存库。暂存库可以在任意时刻将任意一个库中的数据包发送到接收端。同一条信息的数据包必须按顺序连续地被接收端接收。求暂存库的最小大小。

算法关键字

枚举 模拟

算法讨论

先枚举信息被接收端接收的顺序，后用链表模拟暂存库中数据包的出入，取暂存库大小的最小值输出。

时空复杂度

时间：O(n!m)

空间：O(nm)

2007 I - Water Tanks

题目大意

n（n<=10）个水箱排成一行，第一个水箱不封顶，其余的封顶，所有水箱都有各自的高度，水箱底面积都为 1，底面都在同一水平面上。相邻两个水箱之间由水管连接，水管高度从左到右递增。水管允许水或空气自由流动，但体积忽略不计。空气可以被压缩。求忘第一个水箱中注水直到水溢出时，所有水箱内共有多少体积的水。

算法关键字

计算几何

算法讨论

除了第一个和最后一个水箱，其余水箱的水面高度有 3 种情况：在底面与左面水管之间、在左面水管与右面水管之间、在右面水管与顶面之间。水一定会先涨到第 i 个水箱的右管道，再向 i+1 号水箱注水，若 i+1 号水箱的水也涨到了 i 的右管道，i 的水面才会继续往上涨。所以，枚举最后一个有水的水箱，再依次假设水面处在哪一段，求出当 1 号水箱注满水时最后一个水箱的水面高度后判断是否与假设相符。有了最后一个水箱的水面高度后，就可以确定其余所有水箱的水面高度。

时空复杂度

时间：$O(n\log\frac{1}{eps})$

空间：O(n)

2007 J - Tunnels

题目大意

你的基地中有 R（R<=50）个路口和 T（T<=1000）条连接两个路口的无向通道。一个间谍从 1 号路口出发，前往 0 号路口。当间谍出现在某个路口时，你可以选择炸毁某些通道。求在最坏情况下需要炸毁多少条通道，才能使间谍无法到达 0 号路口。

算法关键字

最小割

算法讨论

先分别以 1 到 R 每个点为源、0 号点为汇做一次最小割。记录 i 号点为源时最小割为 g[i]，显然若在间谍出现在 i 点时一次性炸毁所有要炸的通道，答案即为 g[i]。然而为了使答案最优，我们需要分次炸毁通道来控制间谍的行动。我们删去 g[i]最小的所有点（设这个最小值为 gm），以剩下的点为源再分别做最小割，记 i 号点新的最小割为 g’[i]，则当间谍出现在 i 点时，要么仍旧选择炸毁原方案的通道，答案为 g[i]，要么选择炸毁第二次最小割中的割边，使间谍只能进入第一次最小割为 gm 的点，到时候再困住他，答案为 g’[i]+gm。将 $\min\{g[i], g'[i] + gm\}$作为 i 号点新的答案。重复删点和做最小割的操作，直到所有的点都被删掉。最终答案即为 g[1]。

时空复杂度

时间：$O(R^4T)$

空间：O(R+T)

2008 A - Air Conditioning Machinery

题目大意

你需要在 X*Y*Z（X,Y,Z<=20）的长方体内装 L 形管道。每段管道占据 4 个格子，且两端有开口，管道和管道之间可以首尾相连，但不能交叉，也不能超出长方体边界。给出管道入口和出口的坐标和方向，求最少用几段管道可以连通入口和出口。你最多只能用 6 段管道。

算法关键字

搜索

算法讨论

因为数据范围较小，可以直接搜索，搜索时记录下当前管道末端的坐标和朝向，当前有哪些格子已被占据，当前已用几段管道即可。

时空复杂度

时间：$O(8^6)$
空间：O(X*Y*Z)

2008 B - Always an Integer

题目大意

给定一个自变量为 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。

算法关键字

数论

算法讨论

一个关于 x 的多项式 f(x)恒被 d 整除，等价于：

1.f(1)被 d 整除；

2.对于任意的正整数 x，f(x+1)-f(x)被 d 整除。

设 f(x+1)-f(x)=f'(x)，易知 f'(x)的次数 p 一定小于 f(x)，于是在判断了 f(1)是否被 d 整除后，原问题转化为判断 f'(x)是否恒被 d 整除。由归纳法得出，我们只需判断 f(1),f'(1),f''(1)……是否能被 d 整除即可。而 f(1)=f(1)，f'(1)=f(2)-f(1)，f''(1)=f(3)-2f(2)+f(1)……所以，判断对于任意的 1<=x<=p，f(x)是否能被 d 整除，等价于判断 f(1),f'(1),f''(1)……是否都被 d 整除。

时空复杂度

时间：O(p+d)

空间：O(p+d)

2008 F - Glenbow Museum

题目大意

一座博物馆的俯视图是一个直角多边形，一个'R'和'O'组成的字符串按顺时针顺序描述了多边形的每个内角，'R'表示 90°，'O'表示 270°。多边形每条边的长度是不确定的。若博物馆内存在一点，使得站在该点的观察者能看到整个博物馆，则这种设计博物馆是安全的。求有多少个长度为 n（n<=1000）的 RO 字符串对应了至少一种安全的博物馆设计。

算法关键字

组合计数

算法讨论

可以直观地得出：每种安全的博物馆设计必然对应一个凸的直角多边形，若以观察者所在点为原点建立坐标系，则直角多边形每个象限内的那部分一定是阶梯形的，对应的字符串一定形如“RORO……OR”。可以枚举第一象限内有几个顶点，另外三个象限内的方案数用组合数求解。设 m=$\frac{n-4}{2}$，答案为$\sum_{i=0}^{m}(i\times 2+1)\times C(m-i+2,2)$.

时空复杂度

时间：O(n)

空间：O(1)

2008 G - Net Loss

题目大意

给出一个 n（n<=10）次函数 p(x)，求一个连续的、由两条线段 $y = a1x + a0$ 以及 $y = b1x + b0$ 组成的函数 g(x)，使得$\int_{-1}^{1}(\mathrm{p}(\mathrm{x})-\mathrm{d}(\mathrm{x}))^2\,\mathrm{dx}$最小。两条线段的交点的横坐标必须是一个给定的值 c。

算法关键字

三分

算法讨论

当两条线段的交点的纵坐标也确定时，交点两边的两段函数是相互独立的，每段的积分是关于线段的斜率的一个二次函数，可以用三分求出最小值。于是，可以外层三分线段交点的纵坐标，内层三分斜率来求出最优解。

时空复杂度

时间：$O(n^2\log^2\frac{1}{eps})$

空间：O(n)

2008 I - Password Suspects

题目大意

已知一个长度为 n（n<=25）的字符串的 m（m<=10）个子串，字符串仅由小写字母构成，求有多少种可能的原串。若可能的原串不超过 42 个，按字典序从小到大输出它们。

算法关键字

AC 自动机 动态规划

算法讨论

为方便处理，去除重复和被其它串包含的子串，建立 AC 自动机，用状态 f[i][j][sta]表示确定了原串的前 i 为，最后一位对应 AC 自动机上的节点 j，已经包含的子串的集合为 sta 的方案数。转移时枚举下一位填什么字母，因为要输出方案，所以要记录一个状态的前导状态。由于转移不到最终状态的无效状态可能很多，所以只有当一个状态的前导状态不超过 42 个时才继续添加前导状态。

时空复杂度

时间：$O(26n^2m2^m)$
空间：$O(42n^2m2^m)$

2008 J - The Sky is the Limit

题目大意

二维平面内有 n（n<=100）个等腰三角形，三角形的底边都在 x 轴上。求所有三角形并起来组成的图形除底边外的轮廓线长度。

算法关键字

计算几何 扫描线

算法讨论

先求出所有三角形之间形成的交点，过每个三角形的顶点和三角形之间的交点作一条竖直的扫描线，相邻两条扫描线之间一定是若干条只在端点相交的线段，需要计入答案的是中点 y 坐标最大的那条线段。

时空复杂度

时间：$O(n^3)$
空间：$O(n^2)$

2009 A - A Careful Approach

题目大意

有 n（n<=8）架飞机需要降落，每架飞机有一个可降落时段[x[i],y[i]]（x[i],y[i]的最大值T<=1440），安排一种降落方案，使得相邻两架飞机降落时刻的间隔的最小值最大，并输出这个最大值。

算法关键字

二分 搜索

算法讨论

二分最小间隔后用搜索判断是否存在可行方案。

时空复杂度

时间：O(n!logT)
空间：O(n)

2009 F - Deer-Proof Fence

题目大意

在二维平面内有 n（n<=9）棵树，每棵树视为一个点。要求建立若干个围栏，将所有树围起来，且使得围栏上任意一点到每棵树的距离>=m。求围栏的最短长度。

算法关键字

计算几何 状态压缩动态规划

算法讨论

用 f[sta]表示将 sta 集合内的树围起来的最短总长度。枚举下一个围栏围住哪些点，求出它们构成的凸包的周长，则需要的围栏长度为凸包周长+2πm。因为围栏相交答案肯定不优，所以可以不考虑相交的情况。

时空复杂度

时间：$O(3^n n\log n)$
空间：$O(2^n)$

2010 B - Barcodes

题目大意

条形码是由黑白相间的条纹的粗细来存储信息的。粗的条纹宽度是细的条纹的两倍。可以用一个二进制串表示条形码条纹的粗细。每 5 个二进制位表示一个字符，它们可能是 1 到 9，‘-’或表示开头和结尾的特殊字符。原串在末尾加上两位校验字符，再在头尾各加一个特殊字符即为需要编码的字符串。给出一个有 n（n<=150）个条纹的条形码每条条纹的宽度，求解码后的字符串，或判断无法解码、或判断可以正确解码但校验字符不正确。给出的条形码可能是反向的，条形码宽度可能有 5%的偏差，宽度<=250。

算法关键字

模拟

算法讨论

首先判断条形码的条纹条数是否正确。为了得出条形码的标准宽度，需要先枚举细条纹的宽度，判断每条条纹是否能正确识别。如果存在零种或两种以上识别方案则判断无法解码。之后枚举条形码是正向的还是反向的，判断是否存在无法解读的编码、是否存在头尾特殊字符、校验字符是否正确即可。

时空复杂度

时间：O(n)
空间：O(n)

2010 C - Tracking Bio-bots

题目大意

在一个可以描述为一个 m×n（$m,n<=10^6$）网格的房间里有一个机器人，房间内有 w（w<=1000）个障碍物，每个障碍物是一面东西走向的墙。出口在房间的右上角，机器人只能向北或向东走，求房间内有多少个格子，使得机器人从这个格子出发无法到达出口。

算法关键字

离散化 宽度优先搜索

算法讨论

将障碍物的坐标离散化后，整个网格最大为 2000*2000，从出口开始倒着 BFS 一遍，求出哪些格子无法被遍历到，统计这些格子的实际大小总和后输出。

时空复杂度

时间：$O(n+m+w^2)$
空间：$O(n+m+w^2)$

2010 D - Castles

题目大意

一个树形结构的城堡群中有 n（n<=100）个城堡和 n-1 条无向道路，每个城堡有 3 个属性：攻打该城堡需要的士兵数、攻下该城堡后死亡的士兵数、攻下该城堡后需要留守在城堡内的士兵数。第一个攻打的城堡可以任意选择，之后可以经过已攻下的城堡移动到其它城堡继续攻打。求攻下全部城堡至少需要多少士兵。

算法关键字

枚举 贪心

算法讨论

首先枚举第一个攻打的城堡是哪一个。对于每个城堡求出两个值：e[i]表示攻下以第 i 个点为根的子树中的所有城堡需要多少士兵，d[i]表示会损失多少士兵（包括死亡的和占领的）。在攻下了 i 点后，按 e[k]-d[k]的值从大到小攻打每个子树，逆推一遍就可以得出 e[i]和 d[i]。

时空复杂度

时间：$O(n^2 \log n)$
空间：$O(n)$

2010 G - The Islands

题目大意

二维平面内有 n（n<=100）个岛，每个岛都视为一个点，且横坐标互不相同。选择一条环路正好经过 n 个岛，要求这条环路可以分成两段，使得第一段经过的岛横坐标递增，第二段经过的岛横坐标递减，且两个特殊岛 b1 和 b2 不处于同一段中。输出最短环路总长和方案。

算法关键字

动态规划

算法讨论

想象有两个人同时从最左端的岛出发，选择两条无公共点的路径到达最右端的岛，恰好经过 n 个点，且一个人经过 b1，另一个人经过 b2。用 f[i][j][k]表示第一个人处于 i，第二个人处于 j，第 k 个人经过 b1 的最短走过距离。下一步一定是某个人走到 max(i,j)+1 号点，枚举由哪个人走到进行转移，特判下一个点是 b2 的情况。

时空复杂度

时间：$O(n^2)$
空间：$O(n^2)$

2010 J - Sharing Chocolate

题目大意

将一块 X*Y（X,Y<=100）的巧克力分成 n（n<=15）份，要求每份都是一个矩形，边长都为整数，第 i 份的面积为 a[i]。每次分割可以选择将某小块巧克力竖直地或水平地一分为二。问是否存在一种分割方案符合要求。

算法关键字

状态压缩动态规划

算法讨论

用状态 f[sta][i]表示一块宽为 i 的巧克力能否分割成 sta 中的几块巧克力，巧克力的长可以通过总面积和宽求出。枚举此次操作是水平分割还是垂直分割，再枚举 sta 的子集表示其中一小块包含哪几份，若存在一种方案使两小块都能完成分割，则 f[sta][i]值为 TRUE。

时空复杂度

时间：$O(n*3^n)$

空间：$O(n*2^n)$

2010 K - Paperweight

题目大意

两个四面体拥有一个公共面，拼成一个六面体纸镇，纸镇内部有一块芯片，视为一个点。给出六面体顶点和芯片的坐标，求出当纸镇“平稳”地放在地面上时，芯片离地面的最近和最远距离。“平稳”指纸镇的重心从原位置向任意方向移动 0.2 个单位长度后，整个纸镇仍然保持静止。

算法关键字

枚举 计算几何

算法讨论

因为至少有 3 个顶点接触地面，所以可以枚举哪 3 个点一定在地面上。此时判断是否有第 4 个点也在地面上，若没有，则需判断另外 2 个点是否在这 3 个点构成的平面的同侧。重心会在一个球面上任意移动，投影到地面上是一个半径为 0.2 的圆。判断一种放置方法是否“平稳”，需要判断整个圆是否都在地面上的顶点构成的凸包内，即：先判断圆心是否落在凸包内，再判断圆心到凸包每条边的距离是否都不小于 0.2。

纸镇是由两个四面体 DABC 和 EABC 拼成的，设点 A 的坐标表示为(Ax,Ay,Az)，则四面体 DABC 的重心的 x 坐标 G1x= $\frac{Ax+Bx+Cx+Dx}{4}$，y,z 坐标类似；设四面体 EABC 的重心为 G2，整个纸镇的重心的 x 坐标 Gx= $\frac{V_{DABC}\times G1x+V_{EABC}\times G2x}{V_{DABC}+V_{EABC}}$，y,z 坐标类似。

本题中可以利用三维向量的混合积，判断点在平面的哪一侧，判断点是否在已知平面上，求四面体的体积等。

设 ***OA***=(x1,y1,z1),***OB***=(x2,y2,z2),***OC***=(x3,y3,z3)，***OA***,***OB***,***OC*** 的混合积（记为$[\boldsymbol{OA}\ \boldsymbol{OB}\ \boldsymbol{OC}]$）等于 $\begin{vmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{vmatrix}$.

四面体 OABC 的体积 V_{OABC}= $\frac{1}{6}\times|[\boldsymbol{OA}\ \boldsymbol{OB}\ \boldsymbol{OC}]|$.

O,P 两点在平面 ABC 的异面等价于$[\boldsymbol{OA}\ \boldsymbol{OB}\ \boldsymbol{OC}]$ 与 $[\boldsymbol{PA}\ \boldsymbol{PB}\ \boldsymbol{PC}]$ 异号.

求点 O 到平面 ABC 的垂足 H 的坐标，可根据以下条件联立方程组求解：

$$\begin{cases} OH\perp AB \\ OH\perp AC \\ V_{HABC}=0 \end{cases}$$

点 O 到平面 ABC 的距离= $\frac{3\times V_{OABC}}{S_{\triangle ABC}}$，其中 $S_{\triangle ABC}$ 可用海伦公式求解.

点 O 到直线 AB 的距离= $\frac{2\times S_{\triangle OAB}}{AB}$.

时空复杂度

时间：$O(n*3^n)$
空间：$O(n*2^n)$

2011 D - Chips Challenge

题目大意

在一块 n*n（n<=40）的芯片上，某些格子上已有组件，某些格子可以选择装上组件，其余的不能装组件。要求第 i 行的组件数和第 i 列的组件数相同，且每行、每列的组件数不超过总组件数的 A/B。求最多能再装多少组件。

算法关键字

枚举 费用流

算法讨论

如果将格子(i,j)上的组件视为一条 i 到 j 的有向边，因为要满足第 i 行和第 i 列组件数相同，所有边一定构成了若干个环，且每条边只属于一个环。于是此题可以转化为网络流模型。首先枚举每行每列最多装的组件数 m，将每行每列都视为一个点，在原芯片上若第 i 行第 j 列可以装组件，则 i 向 j'连流量 1 费用 1 的边；对于每个 i，源向 i 连流量 m 费用 0 的边，i'向汇连流量 m 费用 0 的边，i 向 i'连流量 m 费用 0 的边。对于已有组件的格子(i,j')，直接在源到 i、j'到汇的边上扣去 1 的流量。因为要满足流量平衡，所以若存在合法方案，所有费用 1 的被流过的边一定构成了若干个环。求一遍最大费用最大流，费用即为最多能装上的组件数。若 m<=组件数*A/B，且费用流给出的方案满足第 i 行和第 i 列组件数相同，则该方案为一组可行方案。取 m 最大的可行方案的费用输出。

时空复杂度

时间：$O(n^5)$
空间：$O(n^2)$

2011 F - Machineworks

题目大意

一家公司有初始资金 C（$C<=10^9$），在 D（$D<=10^9$）天内通过购买机器进行获利。共有 n（$n<=10^5$）台机器，第 i 台机器只可以在第 di 天购买，价格为 pi，卖出价为 ri，每天能为公司带来 gi 的利润。公司只能同时拥有一台机器，每台机器可以选择在任意一天卖掉。求 D 天后公司拥有的最大资金数。

算法关键字

动态规划 线段树

算法讨论

每台机器给公司带来的获利是关于时间的一次函数。将机器按 di 从小到大排序，第 i 台机器会在某台机器 j 工作到第 di-1 天，在第 di 天卖掉 j 后被买入。可以用线段树维护之前出现的所有一次函数在 x=di-1 时的最大值，若当天的资金足够购买机器 i，则在线段树中插入一个新的一次函数，并更新答案。

时空复杂度

时间：O(nlogn)
空间：O(n)

2011 H - Mining Your Own Business

题目大意

一个矿井中有 M 个挖矿点和 N（N<=50000）条无向通道。整个矿井是连通的。你需要在某些挖矿点建造安全竖井，使得任意一个挖矿点坍塌后，其它挖矿点的工人都能到达某个安全竖井。求最少建造几个安全竖井，以及在满足最少的前提下有多少种方案。

算法关键字

割点 组合计数

算法讨论

先用 Tarjan 算法求出割点，若一个联通块只与一个割点相连，则当割点坍塌后这个联通块内就需要一个安全矿井让工人逃生。所以，最少建造数即为只与一个割点相连的连通块数，方案数为这些联通块大小的乘积。需要特判整个图是一个联通块的情况，只需要在任意 2 个点建安全竖井，方案数为 C(n,2)。

时空复杂度

时间：O(N)
空间：O(N)

2011 I - Mummy Madness

题目大意

二维平面内有 n（$n<=10^5$）个木乃伊，每一单位时间内你和木乃伊都可以走到周围 8 个格子之一，或者选择不走，若某一时刻你的位置和某一只木乃伊重合了，你就被抓住了。求你最多能坚持多长时间不被抓住，或者判断永远不会被抓住。初始坐标绝对值 $L<=10^9$。

算法关键字

二分

算法讨论

在时刻 t 你和木乃伊可能所处的位置都构成一个以初始位置为中心，边长为 2t+1 的正方形。二分答案后问题转化为判断一个正方形是否被其它正方形完全覆盖。依次处理每个正方形，若发现它部分覆盖了原正方形，则将原正方形分裂成 2 个小矩形，或者剩下一个较小的矩形，继续被其它正方形覆盖。因为只有当正方形的一个角落在被覆盖的矩形内时，才会把矩形分割成 2 个较小的矩形，且覆盖正方形和原正方形是等大的，所以分裂出的矩形总数量是 O(n)级别的。

时空复杂度

时间：O(nlogL)

空间：O(n)

2011 K - Trash Removal

题目大意

将一个 n（n<=100）个顶点的多边形平移过一个管道，求管道的最小宽度。

算法关键字

计算几何

算法讨论

给出的多边形的答案与该多边形所有顶点构成的凸包的答案相同。将凸包平移过管道时，必有一条边是与管道平行的，否则肯定不优。于是可以枚举哪条边与管道平行，求出与这条边最远的顶点的距离，再在所有距离中取最小值即为答案。

时空复杂度

时间：O(nlogn)
空间：O(n)

2012 B - Curvy Little Bottles

题目大意

一个瓶子过底面中心垂线的截面的一侧可用一个 n（n<=20）次函数 f(x)表示，瓶底对应 x=L 处，瓶口对应 x=R 处。向瓶中注水，每注 Inc 体积的水就在水平线处作一标记。求瓶子的容积，并输出所有标记到瓶底的高度（超过 8 个标记只输出前 8 个）。

算法关键字

积分

算法讨论

易知瓶子从高度 L 到高度 R 的部分的容积为$\int_L^R \pi f^2(x)\,dx$，于是从瓶底开始，二分下一个标记的高度后用公式判断容积与 Inc 的大小关系，即可依次求出每个标记的高度。

时空复杂度

时间：O(nlog$\frac{1}{eps}$)

空间：O(n)

2012 C - Bus Tour

题目大意

有 n（n<=20）个点，m 条无向边。你需要开车从 1 号点出发，经过 2~n-1 号点接游客，一起去 n 号点参观，再将游客送回 2~n-1 号点，最后返回 1 号点。要求前$\left\lfloor\frac{n-2}{2}\right\rfloor$个上车的游客返回时也是前$\left\lfloor\frac{n-2}{2}\right\rfloor$个下车的。求最小的总行驶距离。

算法关键字

枚举 状态压缩动态规划

算法讨论

因为边是无向的，最短路是可逆的，所以可以用状压 DP 预处理 f[i][sta]表示从起点出发，经过的点用二进制表示为 sta，最后停在 i 点的最短距离；g[i][sta]表示其他同上但是从终点出发的最短距离。之后枚举前一半上车的人为哪些、前一半人中最后上车的人和后一半中第一个上车的人，取最优答案输出。

时空复杂度

时间：$O(n^2*2^n)$
空间：$O(n*2^n)$

2012 E - Infiltration

题目大意

在一张有向图中有 n（n<=75）个点，每两个点之间有且仅有一条边。选定一个点可以控制这个点本身和它的出度指向的点。求最少选定几个点可以控制所有点。

算法关键字

搜索

算法讨论

首先明确，任何情况下答案都不会超过 6。n 个点的有向图内两两都连有边，所以总出度数一定是$\left\lfloor\frac{n(n-1)}{2}\right\rfloor$，出度最多的点出度数一定不少于$\left\lfloor\frac{n-1}{2}\right\rfloor$，否则与总出度数$\left\lfloor\frac{n(n-1)}{2}\right\rfloor$产生矛盾。所以，每次一定能控制至少一半的点。由于答案上界很小，可以直接搜索。

时空复杂度

时间：$O(n*\sum_{i=1}^{5}C(n,i))$
空间：$O(n^2)$

2013 A - Self-Assembly

题目大意

给定一个分子的集合，判断是否能构成无限大的结构体。集合中有 n（n<=40000）种分子，每种分子被描述为一个正方形，每条边编号有一个大写字母和一个正负号，或是’00’。只有大写字母相同且正负号相反的边可以贴合。分子可以旋转或翻转，分子之间不能重叠。

算法关键字

图论

算法讨论

由于一个无限大的结构体内必然包含一条无限长的链，所以只需判断判断能否构成无限长的链。因为分子可以翻转或旋转，所以必定存在一种连接方式使得分子不重叠，可以不予考虑。实际不同的连接口只有 52 种，所以可以将连接口看成点，分子看成边，构出一个最多 52 个点的有向图，判断是否存在环，如果存在则说明可以构成无限大的结构。

时空复杂度

时间：O(n)
空间：O(1)

2013 B - Hey, Better Bettor

题目大意

有一种赌博规则如下：每局赌资为 1 元，获胜可以得到 2 元。若某一时刻你的资金比开始时少了 k 元，则可以拿回 k 元的 x%的补偿，但只能拿一次。给出赌局的胜率 p，求在最优策略下期望赚多少钱。

算法关键字

数学期望

算法讨论

赌博的策略可以按如下形式描述：要么在输了 a 元之后结束，要么在赢了 b 元之后结束。只要算出赢 b 元的概率，就可以得出这个策略的期望获利。

假设初始资金为 0，设 $f(x)=\left(\frac{1-p}{p}\right)^{x}$，第 n 局结束时手头的资金为 x_n，则 $f(x_n)=\left(\frac{1-p}{p}\right)^{x_n}$，$f(x_{n+1})$关于 $f(x_n)$的条件期望$= p\times\left(\frac{1-p}{p}\right)^{x_n+1}+(1-p)\times\left(\frac{1-p}{p}\right)^{x_n-1}=\left(\frac{1-p}{p}\right)^{x_n}$，即：$f(x_i)$的条件期望是定值。

因为赌博结果只有赢 b 元和输 a 元两种，设赢 b 元的概率为 P，则 有等式$f(0)=P\times f(b)+(1-P)\times f(-a)$，解得$P=\frac{1-\left(\frac{1-p}{p}\right)^{-a}}{\left(\frac{1-p}{p}\right)^{b}-\left(\frac{1-p}{p}\right)^{-a}}$，最终期望获利即为$P\times b-(1-P)\times(1-x\%)\times a$。我们只要枚举 a 和 b 的值即可求出最大期望获利。枚举时，可以先枚举 a 和 b 的最高位，确定一个较优的解之后继续枚举 a 和 b 的低位，直到符合精度要求。

时空复杂度

时间：$O(\log^2\frac{1}{eps})$

空间：$O(1)$

2013 C - Surely You Congest

题目大意

一张有 n（n<=25000）个点 m（m<=50000）条边的带权无向图，有 c（c<=1000）辆车需要从各自的起点同时出发前往 1 号点。两辆车不能在同一时刻沿同一方向驶上同一条边，否则会发生拥堵。每辆车都必须选择一条最短路行进。求有几辆车能在满足上述条件的前提下到达 1 号点。

算法关键字

最短路 最大流

算法讨论

先求出 1 号点到其余点的最短路。因为最短路长度不同的两辆车一定不会发生拥堵，所以按 1 号点到起点的最短路划分层次进行处理。对于同一层的点，引入汇点 T，所有起点向 T 连流量为起点上车的数量的边。原图中的无向边看做有向边，将可能在最短路上的边流量设为 1，其余设为 0。求出 1 号点到 T 的最大流即为当前层中能不发生拥堵到达 1 号点的车的数量。统计每层的和即为最终答案。

时空复杂度

时间：O(mc)

空间：O(n+m+c)

2013 D - Factors

题目大意

将一个数分解质因数后有若干种表示方法，如 20=5*2*2=2*5*2=2*2*5，则 20 有 3 种表示方法。给出 n（$n<2^{63}$），求最小的 K 使得 K 分解质因数后有 n 种表示方法。保证 $K<2^{63}$，数据组数 T<=1000。

算法关键字

搜索

算法讨论

若一个数分解质因数后为 $p1^{k1}p2^{k2}\cdots pm^{km}$，则这个数有$\frac{(k1+k2+\cdots+km)!}{k1!k2!\cdots km!}$种表示方法。原问题即是求最小的 K，使得 $K=2^{k1}3^{k2}5^{k3}7^{k4}\cdots$，$\frac{(k1+k2+\cdots+km)!}{k1!k2!\cdots km!}=n$，且 k1>=k2>=k3……。因为保证了 K 一定小于 2^{63}，所以可以搜索出所有满足上述条件的 K，求出其表示方法数 n 后列出一张表，询问时直接查询出给定的 n 值中 K 值最小的一条输出。

时空复杂度

时间：O(T)

空间：O(1)

2013 E - Harvard

题目大意

将 m（m<=13）个变量存储在 b（b<=13）个内存库中，每个内存库有 s（s<=13）个栏位。0 号内存库中的变量可以用一个指令直接访问，访问其余的内存库，如果当前 BSR（内存库选择寄存器）不是指向这个内存库的话，需要先用一个指令设定 BSR 的指向，再用一个指令访问内存。给出一个带有循环的访问序列，长度 L<=1000，问最少需要几条指令才能顺次完成所有访问。

算法关键字

搜索

算法讨论

本题中心思想是搜索每个变量存储在哪个位置，加上一些剪枝能大幅度提高程序运行效率：

1、0 号内存库必须填满（除非总变量数不够）；

2、若一个变量要被分配到一个空的内存库中，只选编号最小的那一个；

3、若方案确定后某两个内存库中的变量个数和小于一个内存库的容量，则舍弃这一方案，因为将这两个内存库合并后答案不会变差。

实现时，可以先预处理 f[sta][i][j]表示 0 号内存库中存储的变量为 sta，将这些变量从操作序列中删去后，有多少次先访问变量 i 紧接着再访问变量 j。之后搜索出方案后就可以在 $O(m^2)$的时间内求出这个方案的答案。

时空复杂度

时间：$O(m^{b+2})$

空间：$O(L+m^2)$

2013 F - Low Power

题目大意

有 n 个机器，每个机器有 2 个芯片，每个芯片可以放 k 个电池。每个芯片能量是 k 个电池的能量的最小值。两个芯片的能量之差越小，这个机器就工作的越好。现在有 2nk 个电池，已知它们的能量，我们要把它们放在 n 个机器上的芯片上，使得所有机器的能量之差的最大值最小。

算法关键字

二分 贪心

算法讨论

先将 p[]从小到大排序。一台机器两块芯片中影响答案的那两块电池在排好序的序列中一定是相邻的，否则肯定不优。二分答案 ans 后，每次贪心地选择将哪两块电池作为芯片的最小能量的电池。设已经处理了 i 台机器，则第 i+1 台机器能量最小的那块电池可以在 1~i*2k+1 中选择，每次选择最靠前的能量差不大于 ans 的相邻两节电池即可。

时空复杂度

时间：O(nklog(nk))
空间：O(nk)

2013 H -Матрёшка

题目大意

n（n<=500）个俄罗斯套娃排成一行，每个套娃有一个尺寸（不超过 500），尺寸小的套娃可以放在尺寸大的套娃内部。每次可以合并相邻的两个套娃，代价为需要拆开的套娃数量。一个完好的套娃指若干个标号连续且最小标号为 1 的套娃组成的套娃。求最少花多少代价可以将这些套娃合并成若干个完好的套娃。

算法关键字

区间动态规划

算法讨论

先预处理 i 到 j 有无出现重复的套娃 ok[i][j]， i 到 j 的最小尺寸 min[i][j]，i 到 j 的最大尺寸 max[i][j]，i 到 j 尺寸小于 k 的套娃数量 les[i][j][k]。之后进行区间 DP，合并两个区间(i,k)与(k+1,j)时，设 min[i][k]<min[k+1][j]，则合并代价为 j-i+1-les[i][k][min[k+1][j]]。最后再进行一次 dp，g[i]表示将前 i 个套娃合并成若干完好的套娃的最小代价，转移时若 max[i][j]==j-i+1 且 i 到 j 没有重复的套娃，则说明 i 到 j 可以合并成一个完好的套娃。

时空复杂度

时间：$O(n^3)$
空间：$O(n^3)$

2013 J - Pollution Solution

题目大意

求一个半圆和一个简单多边形的交的面积。多边形顶点数 n<=100，坐标绝对值<=1500，半圆半径 R<=1000。

算法关键字

计算几何　自适应辛普森算法

算法讨论

直接用自适应辛普森算法计算面积。

时空复杂度

时间：O(nlogn*$\frac{R}{eps}$)

空间：O(n)