

IOI2014中国国家集训队作业2试题泛做

姓名： 徐寅展

Contents

ACM-ICPC World Finals 1998

- A. Crystal Clear
- B. Flight Planning
- C. Lead or Gold
- D. Page Selection by Keyword Matching
- E. Petri Net Simulation
- F. Polygon Intersections
- G. Spatial Structures

ACM-ICPC World Finals 1999

- A. Bee Breeding
- C. A Dicey Problem
- D. The Fortified Forest
- E. Trade on Verwegistan
- H. flooded!

ACM-ICPC World Finals 2000

- A. Abbott's Revenge
- B. According to Bartjens
- C. Cutting Chains
- E. Internet Bandwidth
- F. Page Hopping
- G. Queue and A

ACM-ICPC World Finals 2001

- A. Airport Configuration
- B. Say Cheese
- F. A Major Problem
- H. Professor Monotonic's Network
- I. A Vexing Problem

ACM-ICPC World Finals 2002

- A. Ballons in a Box
- C. Crossing the Desert
- E. Island Hopping
- G. Partitions
- H. Silly Sort

ACM-ICPC World Finals 2003

- B. Light Bulbs
- C. Riding the Bus
- D. Eurodiffusion
- F. Combining Images

G. A Linking Loader
H. A Spy in the Metro
I. The Solar System
J. Toll

ACM-ICPC World Finals 2004

D. Insecure in Prague
E. Intersecting Dates
G. Navigation
H. Tree-Lined Streets
I. Suspense!

ACM-ICPC World Finals 2005

A. Eyeball Benders
C. The Traveling Judges Problem
E. Lots of Sunlight
G. Tiling the Plane
I. Workshops
J. Zones

ACM-ICPC World Finals 2006

C. Ars Longa
D. Bipartite Numbers
E. Bit Compressor
F. Building a Clock
G. Pilgrimage

ACM-ICPC World Finals 2007

A. Consanguine Calculations
I. Water Tanks
J. Tunnels

ACM-ICPC World Finals 2008

A. Air Conditioning Machinery
B. Always an Integer
D. The Hare and the Hounds
F. Glenbow Museum
G. Net Loss
H. Painter
J. The Sky is the Limit

ACM-ICPC World Finals 2009

G. House of Cards
H. The Ministers' Major Mess
I. Struts and Springs

ACM-ICPC World Finals 2010

- B. Barcodes
- D. Castles
- F. Contour Mapping
- H. Rain
- J. Sharing Chocolate
- K. Paperweight

ACM-ICPC World Finals 2011

- A. To Add or to Multiply
- E. Coffee Central
- H. Mining Your Own Business
- I. Mummy Madness

ACM-ICPC World Finals 2012

- A. Asteroid Rangers
- C. Bus Tour
- E. Infiltration
- L. Takeover Wars

ACM-ICPC World Finals 2013

- A. Self-Assembly
- B. Hey,Better Bettor
- D. Factors
- E. Harvard
- F. Low Power
- H. Matryoshka
- J. Pollution Solution
- K. Up a Tree

ACM-ICPC World Finals 1998

试题编号

1998 A

题目名称

Crystal Clear

题目大意

以平面上每一个整点为圆心，都有一个半径为0.5的圆。现在有一个格点 n 边形，如果一个圆被这个 n 边形割，且割这个圆的边不过圆心，就说这个圆被破坏了。求没被破坏的圆在 n 边形内的面积总和。 $1 \leq n \leq 25$ ，坐标范围 $|X| \leq 250$ 。

算法讨论

要被计算的圆有三类：过 n 边形的顶点，不过 n 边形的顶点但过 n 边形的边，在 n 边形内部。那只要枚举所有点，再按类别计算面积就行了。

时空复杂度时间： $O(X^2n)$ 空间： $O(n)$

试题编号1998 *B***题目名称**

Flight Planning

题目大意

一个航线被分为 k 航段。飞机在某一航段内在某一高度飞行时所消耗的燃料是确定的，飞机可行的飞行高度有21个。飞机在两个航段之间切换高度的代价也是确定的。要询问整个航线的最少燃料，以及在燃料最少的情况下，每个航线的飞行高度。如果有多解，则输出字典序最小的解。 $1 \leq k \leq 100$ 。

算法讨论

用 $dp[i][j]$ 表示在第 i 段航段的高度为 j 时，前 i 段航段的最少燃料。答案就是 $\min_j dp[k][j]$ 。由于要输出字典序最少的解，所以要从后往前 dp 。

时空复杂度时间： $O(21k)$ 空间： $O(21^2k)$

试题编号

1998 C

题目名称

Lead or Gold

题目大意

有三种原料 a, b, c 以及 n 种混合物。给出每种混合物中 a, b, c 的比例，问是否能用这些混合物，混合出比例为 $A : B : C$ 的混合物。 $1 \leq n \leq 100$ 。

算法讨论

如果令总量为1，那么 $a + b + c = 1$ ，就可以舍去一维。我们用 (a, b) 表示一个物品。考虑两个物品 (a_1, b_1) 与 (a_2, b_2) ，若将它们看成两个点，那么能由它们组成的混合物就相当于这条线段上的任意一点。更一般的，能由若干个点组成的混合物落在这些点形成的平面凸包内。考虑一个可行凸包，对于凸包上的任意一条有向边，目标点都在它的同一侧或者落在它上面。那么，可以先求出可行的所有有向边，再判断这些有向边能否组成一个环，这个用拓扑排序就可以了。

时空复杂度时间： $O(n^2)$ 空间： $O(n^2)$

试题编号1998 *D***题目名称**

Page Selection by Keyword Matching

题目大意

现在有一些网页以及一些查询。每一个网页和查询都由小于等于8个字符串构成。如果定义一个网页和一个查询的相关度

$$P = \sum_{i=1}^8 \sum_{j=1}^8 (8-i+1)(8-j+1)[S_i = S'_j]$$

其中 S_k 表示网页的第 k 个字符串， S'_k 表示询问的第 k 个字符串。要求对每个询问输出相关度最大的5个网页。 $1 \leq$ 询问个数 n ，网页个数 $m \leq 25$ ， $1 \leq$ 字符串长度 $l \leq 20$ 。

算法讨论

直接暴力匹配就可以了。

时空复杂度

时间： $O(nml)$

空间： $O(ml)$

试题编号1998 *E***题目名称**

Petri Net Simulation

题目大意

有 nt 个事件和 np 个仓库，每个事件是将某些仓库中的令牌都减一，且在另一些仓库中的令牌都加一。现在要进行 nf 次事件，请输出在 x 次事件后无法进行任何事件，或者输出能进行完 nf 次事件。并且输出在结束后每个仓库的令牌数。 $0 < nt, np < 100$, $0 < nf < 1000$ 。

算法讨论

这是一道模拟题。每次随便找一个可以进行的事件进行。如果没有可以进行的事件，则判断结束。

时空复杂度时间: $O(Np \cdot Nt \cdot Nf)$ 空间: $O(Np \cdot Nt)$

试题编号1998 *F***题目名称**

Polygon Intersections

题目大意

给定一个简单 n 边形和一个简单 m 边形，求出所有它们共同占有的平面区域。 $1 \leq n, m \leq 100$ 。

算法讨论

两个多边形相交后，构成的平面区域是一张平面图。平面图的点是多边形的顶点和两个多边形之间的交点，平面图的边还是多边形原来的边。如果一个平面区域同时由两个多边形的边构成，且所有边的方向都是读入的方向，那么这个平面区域就是两个多边形相交的平面区域。

时空复杂度时间： $O(nm \log nm)$ 空间： $O(nm)$

试题编号1998 *G***题目名称**

Spatial Structures

题目大意

一个 $n * n$ 的矩阵，里面只有01。请你将这个转换为四分树，要求当做到一个节点时，该节点包含的数字全部为0或全部为1就停止。或者把一个四分树转化为这样的矩阵。保证 n 是2的幂且 $n \leq 64$ 。

算法讨论

直接模拟即可。

时空复杂度时间: $O(n^2 \log n)$ 空间: $O(n^2)$

ACM-ICPC World Finals 1999

试题编号

1999 A

题目名称

Bee Breeding

题目大意

给出一张六边形的按顺时针编号的图，求出编号 a 到编号 b 格子的最短距离 $a, b \leq 10000$ 。

算法讨论

在平面上，用两个不平行的向量即可表示出所有向量。如果把1号点向2号点的向量作为 \vec{x} 向量，1号点向6号的向量作为 \vec{y} 向量，那么我们可以求出1号点到每个格子的向量。由于每一步可以走 $\pm\vec{x}, \pm\vec{y}, \pm(\vec{x} + \vec{y})$ 。枚举走了多少步 $(\vec{x} + \vec{y})$ ，那么还要走 \vec{x} 与 \vec{y} 的系数的绝对值之和步，得到的最小值就是答案。

时空复杂度时间： $O(a + b)$ 空间： $O(a + b)$

试题编号

1999 C

题目名称

A Dicey Problem

题目大意

给定一张 $n * m$ 的图，图上每个点有一个值 -1 或 $1 \dots 6$ 。一个骰子放在这张图上，当骰子的顶面数字是 x 时，它可以向值为 x 或 -1 的点滚动，请你输出一个合法的移动序列，使骰子回到原地；或者判断无解。 $n, m \leq 10$ 。

算法讨论

$dp(x, y, s, t)$ 表示骰子在 (x, y) 这个点，顶面是 s ，前面是 t 的状态是否可能出现。bfs后倒推输出序列即可。

时空复杂度时间: $O(nm)$ 空间: $O(nm)$

试题编号1999 *D***题目名称**

The Fortified Forest

题目大意

有 n 棵树，每棵树有一个坐标一个价值，以及砍断之后可以作为木料的长度。问最少砍价值为多少的树，能用砍下后的木料把剩下的树围住。 $1 \leq n \leq 15$ 。

算法讨论

枚举砍了哪些树，再求剩下的树的凸包即可。

时空复杂度

时间： $O(2^n n \log n)$

空间： $O(n)$

试题编号1999 *E***题目名称**

Trade on Verweggistan

题目大意

有 w 个公司，每个公司有一些货物。货物从上往下堆，要买下面的货物，只能先买光上面的所有货物。现在给出每个货物的价格，以及你将要卖的售价，求出最大利润，以及买多少个货物可以达到最大利润。 $1 \leq w \leq 50$, $0 \leq$ 每个公司货物数 ≤ 20 。

算法讨论

最大利润就是最大前缀和。将每个公司达到最大利润时的可能货物数用背包并起来，就是总的可能货物数。

时空复杂度时间: $O(wb^2)$ 空间: $O(wb)$

试题编号1999 *H***题目名称**

flooded!

题目大意

有 nm 个区域，每个区域有一个高度。水会先在高度低的区域上升，直到与高的地区一样高，然后会一起上升。现在给出这些区域的总水量，问水的高度，以及水覆盖了多少区域。

算法讨论

先将这 nm 个区域按高度排序，之后可以二分哪些区域是有水覆盖的。知道了之后就相当解一个一元一次方程。

时空复杂度时间: $O(nm \log nm)$ 空间: $O(nm)$

ACM-ICPC World Finals 2000

试题编号

2000 A

题目名称

Abbott' s Revenge

题目大意

有一个迷宫，每个节点规定从某个方向进入后可以往哪个方向走。问从起点到终点的最短路。节点坐标在1到9之间，数据组数 ≤ 1000 。

算法讨论

由于每一步的代价都是1，可以直接bfs，记录每个点的前驱是什么，再递归输出。

时空复杂度

时间： $O(Tn^2)$

空间： $O(n^2)$

试题编号

2000 *B*

题目名称

According to Bartjens

题目大意

有一个长度为 n 的数字串，要在这个数字串中添加+、-、*号，使得整个表达式的值为2000。 $1 \leq n \leq 9$ 。

算法讨论

枚举每两个字符之间的符号是+、-、*或没有符号，之后计算表达式的值。

时空复杂度

时间： $O(n4^n)$

空间： $O(n)$

试题编号2000 *C***题目名称**

Cutting Chains

题目大意

你现在有 n 个链环，它们中的一些相互套在一起。你需要打开并关上最少的链环，使所有链环形成一条链，输出最少的次数。 $1 \leq n \leq 15$ 。

算法讨论

2^n 枚举打开了哪些链环。那么考虑把这些链环去掉后剩下的图形是否是一些链。如果是且链的条数 \leq 被打开的链环数+1，那么可以用打开的链环数更新答案。

时空复杂度时间: $O(n^2 2^n)$ 空间: $O(n^2)$

试题编号2000 E **题目名称**

Internet Bandwidth

题目大意

给定一张 n 个节点， m 条边的图，求 s 到 t 的最大流。 $n \leq 100$, $m \leq \frac{n(n-1)}{2}$, 所有容量 ≤ 1000 。

算法讨论

运用求最大流的任意一种算法都可以解决此问题。

时空复杂度

时间: $O(n^2m)$

空间: $O(n + m)$

试题编号

2000 F

题目名称

Page Hopping

题目大意

给定一张 n 个节点的图，求两两之间最短路的平均值。 $1 \leq n \leq 100$ 。

算法讨论

可以用 *floyed* 解决。

时空复杂度

时间: $O(n^3)$

空间: $O(n^2)$

试题编号

2000 G

题目名称

Queue and A

题目大意

有 n 种不同的主题，每种主题有一个出现的次数 b_i ，已知每次在什么时刻出现。有 p 个小组成员，每个小组人员有一些可以做的主题种类以及做的顺序。每当一个主题出现时，可以让某一个小组人员操作这一主题；某个小组人员操作完成时，也可以操作一个已经出现的主题。问需要多少时间能处理完所有主题。 $1 \leq n \leq 20$, $1 \leq b \leq 1000$, $1 \leq p \leq 5$ 。

算法讨论

让时间慢慢推进，只有在小组人员完成操作和新主题加入的时候才会产生新的操作。那只要在每个关键时间点不断进行操作即可。

时空复杂度时间: $O(n^2bp)$ 空间: $O(n + p)$

ACM-ICPC World Finals 2001

试题编号

2001 A

题目名称

Airport Configuration

题目大意

机场里有一条大街，有很多游客会从街的一边走向街的另一边。街的每一边都有 n 个门。现在有 m 个方案。每一个方案给出这些门两两之间的游客流量，要计算总的客流指数。总的客流指数是每对门之间客流指数的和。一对门之间的客流指数为这对门之间的距离乘这对门之间的游客流量。 $1 \leq n \leq 25$, $1 \leq m \leq 20$ 。

算法讨论

可以对每个方案模拟。

时空复杂度

时间: $O(n^2m)$

空间: $O(n^2)$

试题编号2001 *B***题目名称**

Say Cheese

题目大意

一块奶酪里有 n 个洞，虫子在洞里的速度是无穷大，在奶酪里的速度是0.1。现在它要从起点跑到终点，求最少时间。 $0 \leq n \leq 100$, $-10000 \leq \text{坐标} \leq 10000$, $0 \leq \text{半径} \leq 10000$ 。

算法讨论

由于在洞内的任何位置都是等价的，可以将洞视为点用最短路解决。

时空复杂度时间: $O(n^2)$ 空间: $O(n)$

试题编号2001 *F***题目名称**

A Major Problem

题目大意

给出每个音符的音阶。升调就是音阶加一，降调就是音阶减一。大调音阶由音符开始并且紧跟着连续的“全音-全音-半音-全音-全音-全音-半音”组成。全音是两个音阶，半音是一个音阶。其中每个字母都出现一次，不允许即出现升调又出现降调。现在给出两个大调音阶和一个音符，这个音符在其中一个音阶中在第 x 个位置，求出另一个音阶中第 x 个位置的音符或判断不合法。数据组数不超过100。

算法讨论

枚举这两个大调音阶的第一个字符，然后暴力判断就可以解决。

时空复杂度

时间： $O(T)$

空间： $O(1)$

试题编号

2001 *H*

题目名称

Professor Monotonic's Network

题目大意

给定一个有 n 个输入端的比较网络，其中有 m 个比较器。判断这个网络是否为排序网络，且输出这个网络工作需要的单位时间。 $1 \leq n \leq 12$, $1 \leq m \leq 150$ 。

算法讨论

第二问可以用一次拓扑排序解决。对于第一问，需要用到排序网络的0-1原则：若一个比较网络对于输入端都为0或1的输入能给出正确答案，那么它能对任意输入给出正确答案。

时空复杂度

时间: $O(2^nm + m^2)$

空间: $O(n + m)$

试题编号2001 *I***题目名称**

A Vexing Problem

题目大意

这是一个经典的游戏。一个 $n * m$ 的区域内有一些障碍，一些空格，以及一些标有字母的方块。每次可以向左或向右移动一个方块，方块会遵循重力。标有相同字母的方块碰到一起会自动消失，问最少多少步能消掉所有方块，保证答案小于等于11。 $n, m \leq 10$ 。

算法讨论

这是一道的搜索题。可以用bfs来计算答案。因为枚举量非常大，因此需要用到如下剪枝：1.如果有重复的状态，就不要再做了，这个可以用set或hash实现；2. 如果某时标有某种字母的方块只剩一个，那么这种状态就是不可行的；3.如果某种字母出现的次数是两次，那么这两个方块在水平方向上必然要一步一步走过去才能消掉，这是必须走的步数，如果必须走的步数加已经走的步数大于11，那这种状态就不可能是最优答案了。同时还要注意的一点是，题目中给出的字符串中间可能含有空格。

时空复杂度时间： $O(11^{nm})$ 空间： $O(11^{nm})$

ACM-ICPC World Finals 2002

试题编号

2002 A

题目名称

Ballons in a Box

题目大意

一个长方体盒子里有 n 个气球。每次吹一个气球，可以让它变大，直到碰到其他气球或盒子的六个面，在别的气球里面的气球不能吹。你要找一个吹气球的顺序，使得总的气球的体积最大。有些气球可以不吹。 $n \leq 6$ 。

算法讨论

枚举吹气球的顺序，依次求出气球的半径，计算体积即可。

时空复杂度

时间： $O(n!)$

空间： $O(n)$

试题编号

2002 C

题目名称

Crossing the Desert

题目大意

你需要穿越一个沙漠，沙漠中有 n 片绿洲，一个起点和一个终点。在这些点上你可以拿到无限的水，在起点你可以买到食物，且在这些点上可以囤积食物。走一个单位的路程，你需要花费一个单位的食物和一个单位的水。你有一个负重 tot ，表示身上能带的水和食物的总量的最大值。求出最少需要买多少食物才能从起点走到终点。 $n \leq 20$ 。

算法讨论

用 $f[x]$ 表示最少需要在 x 点囤积多少食物才能走到终点。如果 x 可以走到 y ，那么肯定是先在 x 囤积食物，等食物够量的时候，再一次次搬到 y 。若已经求出 $f[y]$ ，自然可以推出 $f[x]$ ， $f[1]$ 上取整就是答案， $f[n] = 0$ 。由于如果用 $f[x]$ 转移到 $f[y]$ ，那么 $f[y] > f[x]$ ，因此可以用dijkstra算法求最短路。

时空复杂度时间： $O(n^2)$ 空间： $O(n)$

试题编号2002 *E***题目名称**

Island Hopping

题目大意

平面上有 n 个点，两个点之间的距离为欧几里德距离。求出这些点的最小生成树中所有点到1号点的路径上最大边的带权平均数。 $1 \leq n \leq 50$ 。

算法讨论

用任意一种最小生成树算法均可 *AC* 此题。

时空复杂度

时间： $O(n^2)$

空间： $O(n)$

试题编号2002 *G***题目名称**

Partitions

题目大意

矩形的划分是指把一个矩形分成若干个较小，不重叠的子矩形。如果 B 是 A 通过划分它的一个或多个子矩形得来，那么我们说 B 比 A 更精细，或者说 A 比 B 更粗糙。在比 A 和 B 精细的划分中，存在唯一的一种最粗糙的划分，这种划分被称为 A 和 B 的下确界。比 A 和 B 都粗糙的所有划分中最精细的划分称为 A 和 B 的上确界。现在给出两个 $w * h$ 的矩形，求出它们的上确界和下确界。 $1 \leq w, h \leq 20$ 。

算法讨论

如果在一个矩形中，某条极长的边是 A 的子边或 B 的子边，那么这条边是 A, B 下确界，这条边也把原矩形分成了两个子矩形。之后可以递归做两个子矩形。类似的方法可以求上确界。

时空复杂度时间: $O((w + h)wh)$ 空间: $O(wh)$

试题编号2002 *H***题目名称**

Silly Sort

题目大意

有一个长度为 n 的序列，其中每个数都互不相同。每次可以交换任意两个数字，交换的代价为这两个数字之和，求使这个序列升序所需的最小代价，保证序列中的数字互不相同。 $1 \leq n \leq 1000$ 。

算法讨论

我们可以先排序求出 a_x 的最终位置 b_x ，这样可以将原序列拆分为若干个环，若环中某一个元素对应的位置为 x ，它的后继对应的位置为 y ，那么 $b_x = y$ ，每一个环的答案都是可以单独计算的。对于一个环，可以将环里最小的一个 a_x 与其他所有元素换一遍；也可以将全局最小的元素与最小的 a_x 交换，与别的元素都换一遍后，再换回来。把所有环的答案加起来就是最后的答案了。

时空复杂度时间： $O(n \log n)$ 空间： $O(n)$

ACM-ICPC World Finals 2003

试题编号2003 *B***题目名称**

Light Bulbs

题目大意

一共 n 盏灯， n 个开关。第一个和第 n 个开关分别控制第1、2，第 $n-1$ ， n 盏灯。现在要求用最少的次数把初始时的灯开关情况变成目标开关情况。可以用一个二进制数字表示灯的开关情况，第 i 位是1表示第 i 盏灯是开的。输入的是两个数字，表示初始和目标灯的开关情况，这两个数字是二进制序列的十进制表示。数字位数 $l \leq 100$ 。

算法讨论

将读入转化为二进制序列后，枚举第一个开关的拨动情况。那么可以依次判断2到 n 号开关是否要拨动。若判断完第 i 个开关之后，第 i 盏灯是不符合的，那么这时候第 $i+1$ 个开关一定要被拨动，否则就永远是不符合的。再在两种情况中取一个最优值，转化为十进制就是答案。

时空复杂度时间： $O(l^2)$ 空间： $O(l)$

试题编号2003 *C***题目名称**

Riding the Bus

题目大意

我们定义1阶SZ曲线为一个S型。 k 阶SZ曲线为一个9个 $k-1$ 阶SZ曲线按S型拼起来的图形。求一个 n 阶SZ曲线上亮点的距离。 $1 \leq n \leq 8$ 。

算法讨论

对于一个 n 阶SZ曲线中的两个点，我们可以分别求出它们分别属于哪个 $n-1$ 阶SZ曲线，这样就转化为了 $n-1$ 阶的问题。

时空复杂度时间: $O(n)$ 空间: $O(n)$

试题编号2003 *D***题目名称**

Eurodiffusion

题目大意

一个 $w * h$ 的矩形里有若干个城市，每个城市属于一个国家。共有 n 个国家。开始时每个城市有 1000000 个该国家的货币。若某个时刻某个城市有 x 个某国的货币，那么他会送给四周的城市 $\lfloor \frac{x}{1000} \rfloor$ 个这种货币。问每个国家的所有城市最初在哪个时刻拥有所有类型的货币。 $1 \leq w, h \leq 10, 1 \leq n \leq 20$ 。

算法讨论

如果有一个长度为 100 的城市群，初始时只有最左边那个城市有 1000000 个货币，那么按照题目中的方法，所有城市都有货币只需要 645272 个单位时间。因此按照题意模拟即可。

时空复杂度时间: $O(ans \cdot w \cdot h \cdot n)$ 空间: $O(w \cdot h \cdot n)$

试题编号2003 *F***题目名称**

Combining Images

题目大意

一张正方形黑白图片可以压缩为一串二进制编码。如果当前区域所有像素都相同，那么在二进制编码末尾写一个1，再写上当前区域的颜色；否则，在二进制编码后写上0，再依次写上左上角、右上角、左下角、右下角的二进制编码。现在给出两个二进制编码，请给出两张图片取交后的图片的二进制编码。编码长度 $l \leq 100$ 。

算法讨论

直接按照题目意思建立两棵四分树，再进行取交操作即可。

时空复杂度

时间： $O(l)$

空间： $O(l)$

试题编号2003 *G***题目名称**

A Linking Loader

题目大意

有4种语句 $DECZ$ ， D 是定义一个变量， E 是声明要使用这个变量， C 表示要将后面的变量的值覆盖在之后的内存上， Z 表示结束一个模块。内存的大小是 16^4 ， D 的个数 ≤ 100 。

算法讨论

这题是一道模拟题，主要难点在于对题目的理解上，只要把握好细节就行了，实现起来很简单。

时空复杂度时间： $O(\text{操作数} + DE)$ 空间： $O(\text{操作数} + 16^4)$

试题编号2003 *H***题目名称**

A Spy in the Metro

题目大意

在1号车站的暗探Maria 想在时刻 T 的时候与在 n 号车站的间谍会面。车站从1到 n 编号，并呈直线状。给出相邻两个车站之间需要的行驶时间。正方向行驶的车有 m_1 辆，反方向行驶的车有 m_2 辆，我们会给出每辆车的开车时间。Maria可以在车站等待，也可以在车上，现在要使在车站等待的时间最少。 $2 \leq n \leq 50$, $0 \leq T \leq 200$, $1 \leq m_1 \leq 50$, $1 \leq m_2 \leq 50$ 。

算法讨论

用 $dp[i][j]$ 表示在第 i 个时刻，Maria在 j 号车站所需要的最少等待时间。有两种转移：1.停在车站不动；2.乘上一辆车。在某个时刻某个车站有哪些车可以预处理出来。

时空复杂度时间： $O(n(m_1 + m_2 + T))$ 空间： $O(nT)$

试题编号2003 *I***题目名称**

The Solar System

题目大意

给出一个星系中的两个行星轨道的长轴长，短轴长，以及其中一个的运行周期。求出在时刻 t ，另一个行星的坐标。这里行星的运动都符合开普勒行星运动定律：1. 恒星是行星轨道的一个焦点；2. 行星运动时，相等时间内扫过的面积相等；3. 两颗行星的运行周期平方比等于它们的半长轴立方比。

算法讨论

我们可以先求出另一个行星的运行周期 t_2 ，那么就知道在时刻 t 这个行星运行了几个周期。接下来二分圆心角，可以求出扫过的面积。最后由圆心角得到坐标。

时空复杂度时间： $O(\log \frac{1}{\epsilon})$ 空间： $O(1)$

试题编号2003 *J***题目名称**

Toll

题目大意

你从一个地方出发，要把 p 个银匙卖到另一个地方。当你走进一个村庄时，需要缴纳1银匙的税；当你走进一个城镇时，若身上有 x 个银匙，则需要缴纳 $\lceil \frac{x}{20} \rceil$ 个银匙，一共有26个城镇与26个村庄。问在起点至少要带多少银匙。 $1 \leq p \leq 1000$ 。

算法讨论

令 $dist_i$ 表示当前在 i 号点，最少需要多少银匙才能使到达终点时的银匙大于等于 p 个。这个问题可以用最短路解决。 $dist_S$ 就是答案。

时空复杂度时间: $O(26^2)$ 空间: $O(26^2)$

ACM-ICPC World Finals 2004

试题编号2004 *D***题目名称**

Insecure in Prague

题目大意

有一种对字符串的加密算法。假设加密前的字符串长度为 n ，加密后的字符串长度为 m ，要求 $m \geq 2n$ 。每一次加密算法还需要四个参数 s, i, t, j ， $0 \leq s, i, t, j < m$ ， $i < j$ 。编码方式是这样的：将加密串的第 s 个字符变为原串的第一个字符，接下去每次跳过 i 个空格，在第 $i + 1$ 个空格处填上下一个字符，直到把整个字符串填完。接下去以 t 作为新的起点，以 j 作为新的步长，再添加一边原串。如果还有位置没有填的，把那些位置随机填上一些字符。现在告诉你目标串，请你输出最长的可能的原串。 $1 \leq m \leq 40$ 。

算法讨论

预处理出若字符串长度为 x ，步长为 y ，第 z 次填的字符的位置在哪里。枚举 n, s, i, t, j ，由于有了预处理，可以在 $O(m)$ 的时间内得到原串，判断是否相同即可。实现过程需要注意常数优化。

时空复杂度时间： $O(m^6)$ 空间： $O(m^3)$

试题编号2004 *E***题目名称**

Intersecting Dates

题目大意

给出 n 个已知日期段， m 个需要日期段。在已知日期段中的需要日期段是不需要的。那么需要的日期也是一些日期段，输出这些日期段。 $1 \leq n, m \leq 100$ ，保证日期在1700年到2100年之间。

算法讨论

先预处理出每一天离1700年1月1日的天数，然后就可以暴力做了。

时空复杂度

时间： $O(n + m)$

空间： $O(n + mp)$

试题编号2004 *G***题目名称**

Navigation

题目大意

有一个目标点，一个接收点，和 n 个信号源。告诉你某一时刻接收点接受到的信号的源点和经过时间，求可能的接收点位置，并输出目标点在接收点的什么方向。 $n \leq 10$ 。

算法讨论

这个问题的本质就是求出一些圆的公共交点，直接解方程求就行了。

时空复杂度时间: $O(n)$ 空间: $O(n)$

试题编号2004 *H***题目名称**

Tree-Lined Streets

题目大意

有 n 条街道，你需要在上面种一些树。每两棵树之间的距离至少为50米，树与十字路口的距离至少为25米，问最多能种几棵树。 $n \leq 100$ 。

算法讨论

每次枚举一条大街，找出所有交点，排序后分别处理每一段。由于十字路口向两个方向的限制都有25m，因此不同的段互不影响。

时空复杂度时间: $O(n^2 \log n)$ 空间: $O(n)$

试题编号

2004 I

题目名称

Suspense!

题目大意

有两幢距离为 d 的公寓，要在之间建造一座水平的桥。每层楼有一只猫或一只鸟或什么也没有。若一只猫在高度为 h 窗口,那么它可以跳到高度为 $(h-3, h+0.5)$ 的桥面；反之，也可以从高度为 h 的桥面，跳到高度为 $(h-3, h+0.5)$ 的窗口。桥上有一根桥缆，两端的高度已知，要求桥缆的最低端比桥面高1 米，且猫不能通过桥抓到鸟，求桥缆最长是多少。每层楼高度为3 米，窗户高度为1.5米，窗户离每层楼的地面1米，公寓的层数 $n \leq 50$ 。

算法讨论

由于桥面可以在的高度有无穷多个，对问题的解决带来了很多麻烦。我们以层为单位讨论一些情况，这样可以排除掉不优的解。表格中的 H_i 表示第 i 层的窗台的高度

高度范围	能跳到这个高度的层数	这个高度能跳上的层数
$H_i - 0.5$	i	$i - 1$
$(H_i - 0.5, H_i)$	i	$i - 1, i$
H_i	i	i
$(H_i, H_i + 0.5)$	$i, i + 1$	i
$H_i + 0.5$	$i + 1$	i
$(H_i + 0.5, H_i + 2.5)$	$i + 1$	i

有一个比较明显的结论：桥面越低，缆绳越长。那么 $H_i-0.5$ 比 $(H_i-0.5, H_i)$ 优， H_i 比 $(H_i, H_i+0.5)$ 优， $H_i+0.5$ 比 $(H_i+0.5, H_i+2.5)$ 优。我们已经把桥面可能的高度缩小到了有限个。那么原问题可以转化为如下问题：有一条过原点的抛物线，经过两个点 $(x_1, y_1), (x_2, y_2)$,其中 y_1, y_2 是已知的， $x_2 - x_1 = d$ ，要求该抛物线横坐标在 (x_1, x_2) 那一段的长度。

令抛物线的解析式为 $y = ax^2$

$$\begin{cases} ax_1^2 = y_1 \\ ax_2^2 = y_2 \\ x_2 - x_1 = d \\ x_1 < 0 \\ x_2 > 0 \end{cases}$$

解得

$$a = \frac{\sqrt{y_1} + \sqrt{y_2}}{d}$$

接下来要求的是 $y = ax^2$ 在 x_1 到 x_2 上的长度，亦即

$$\int_{x_1}^{x_2} \sqrt{x'^2 + y'^2} dx = \int_{x_1}^{x_2} \sqrt{1 + 4a^2 x^2} dx$$

由公式可知

$$\int \sqrt{1+4a^2x^2}dx = \frac{x}{2}\sqrt{4a^2x^2+1} + \frac{1}{4a}\ln(2ax + \sqrt{4a^2x^2+1})$$

这样就可以在 $O(1)$ 的时间内求一条抛物线的长度了。

回归本题，只要判断所有可能作为桥面的高度是否可行，若可行再求缆绳长度即可。

时空复杂度

时间： $O(n)$

空间： $O(n)$

ACM-ICPC World Finals 2005

试题编号

2005 A

题目名称

Eyeball Benders

题目大意

给出两组由线段组成的图形，问其中一个是不是另外一个某一部分的放大。保证有一个顶点和一个交点是共有的。线段条数 $n, m \leq 50$ 。

算法讨论

由于有两个参照点，因此可以枚举这两个点，把变换函数算出来，判断一下是否完全相同就好了。

时空复杂度时间: $O(m^3 n(n+m) \log n)$ 空间: $O(n+m)$

试题编号

2005 C

题目名称

The Traveling Judges Problem

题目大意

一张有 nc 个节点的图，其中有 nj 个人，要从 nj 个节点走向 dc 点，当他们的路径重复时只计算一次。请计算最小代价，并输出每个人的路径。存在多种方案时，选择需要访问到的点集合元素最少的一种；仍然存在多种方案时，选择点元素升序排列后字典序最小的一种。 $nc \leq 20, nj \leq 10$ 。

算法讨论

如果没有输出路径的条件，那么这题就是经典的斯坦纳树。有了这个限制。只要在dp时再加一维S，表示走到过的点的集合，那样就可以每次求出最优的S。可以证明每次最优，最终也是最优的。

时空复杂度时间: $O(2^{nc} + 3^{nj} + 2^{nj}nc^2)$ 空间: $O(2^{nc} + 3^{nj}nc)$

试题编号2005 *E***题目名称**

Lots of Sunlight

题目大意

有 n 栋楼房。一间公寓被太阳直射当且仅当整块东侧或西侧的外墙被太阳直射，或太阳位于公寓正上方。每次给出一间公寓，要求输出这间公寓被太阳直射的时间段。 $1 \leq n \leq 100$, 询问次数 $Q \leq 1000$ 。

算法讨论

对于每个询问，将西边和东边的楼房分开处理，求出可行的太阳角度，折算出时间。

时空复杂度时间: $O(nQ)$ 空间: $O(n)$

试题编号2005 *G***题目名称**

Tiling the Plane

题目大意

给出一个边与坐标轴平行的 n 边形，判断它是否能不重不漏地铺满平面。 $n \leq 50$, 周长 $C \leq 50$ 。

算法讨论

由题目中的提示，枚举 $ABCD$ 点，注意这些点肯定在整点处，因为如果不在整点处，就可以向左右移动，来得到一个整点处的方案。只要预处理出某一段与另一段是否相同，判断就是 $O(1)$ 的了。

时空复杂度时间: $O(C^3)$ 空间: $O(C^3)$

试题编号2005 *I***题目名称**

Workshops

题目大意

有 w 个讨论会， r 个房间，每个讨论会有一个人数和持续时间，每个房间也有一个最大人数和最多持续时间。现在要使能在房间里举行的讨论会最多，在满足这个的条件下，使人数最多。 $1 \leq w \leq 1000, 1 \leq r \leq 1000$ 。

算法讨论

按照以下策略贪心：先将所有房间按时间从小到大排序，每次找一个可以放进该房间且人数最多的会议即可。

时空复杂度时间： $O(wr)$ 空间： $O(w + r)$

试题编号2005 *J***题目名称**

Zones

题目大意

有 n 个服务塔，给出每个服务塔负责的人数，以及一些共同包含某些人的服务塔集合。
现要从中选出 m 座服务塔，问最多能负责多少人。 $n \leq 20, m \leq 10$ 。

算法讨论

2^n 枚举服务塔集合，把重复的人数去掉，取最大值。

时空复杂度

时间： $O(2^n m)$

空间： $O(n)$

ACM-ICPC World Finals 2006

试题编号2006 *C***题目名称**

Ars Longa

题目大意

给出一座由 n 个关节和 m 个支杆组成的雕塑。需要判断这个雕塑是否静止，在关节处收到较小的力时是否稳定。 $n \leq 100, m \leq 100$ 。

算法讨论

将每个支杆的力设为 F ，关于每个关节建立一个方程，判断这个方程组是否有解就可以知道是否静止。对于第二问，可以随机每个关节受到的微小的力，重新解一次方程即可。

时空复杂度时间: $O(n^3)$ 空间: $O(n^2)$

试题编号

2006 D

题目名称

Bipartite Numbers

题目大意

定义一个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，并且排列中所有的 s 出现在 t 之前。输入 k ，输出最小的大于 k 且是 k 的倍数的二段数由几个 s ，几个 t 组成，并输出 s, t 。 $1 \leq k < 100000$ 。

算法讨论

我们定义

$$\begin{cases} f(l) = 1 & (l = 1) \\ f(l) = (10f(l-1) + 1) \bmod k & (l > 1) \end{cases}$$

表示 $\underbrace{11 \cdots 1}_{l \uparrow 1} \bmod n$ 的值。由于 $f(l)$ 存在小于等于 k 的循环节，所以答案的长度小于等于 $2k$ 。

考虑一个二段数 $g(s, t, n, m) = \underbrace{ss \cdots s}_{n \uparrow s} \underbrace{tt \cdots t}_{m \uparrow t}$ ， $g(s, t, n, m) \equiv sf(n+m) + (t-s)f(m)$

$(\bmod k)$ 。那么只要枚举 $n+m, s, t$ 并预处理出 f 数组，就能在 $O(1)$ 的时间内得到最优的 n, m ，再与已知答案进行比较就能得到最优答案了。

时空复杂度

时间： $O(10^2 k)$ 空间： $O(10k)$

试题编号2006 *E***题目名称**

Bit Compressor

题目大意

有一种对一个二进制串压缩的算法：对所有极大连续的1，如果将它们写成它们个数的二进制表示能使总长度变短，那么就这样变。给出压缩后长度为 m 的串，以及压缩前串的长度 L ，串中1的个数 n ，问原串是否存在，若存在，是否唯一。 $1 \leq L \leq 65536$ ， $1 \leq n \leq L$ ， $1 \leq m \leq 40$ 。

算法讨论

直接搜索即可通过此题。

时空复杂度

时间： $O(1)$

空间： $O(1)$

试题编号2006 *F***题目名称**

Building a Clock

题目大意

你有一个主转轴和 n 个齿轮，可以使用无限个转轴，使得有一个转轴的速度是24转/天，还有一个是2转/天。 $n \leq 6$ 。

算法讨论

这题是一道搜索题。由于测试点比多，应尽量优化常数。一个比较好的优化就是先搜索符合分针的方案，再搜索时针的方案。

时空复杂度时间: $O(n! \cdot 2^n \cdot n^3)$ 空间: $O(n)$

试题编号

2006 G

题目名称

Pilgrimage

题目大意

有一个旅行队，有四种操作，共 n 个。IN表示加入 k 个人，每个人会支付平均存款；OUT表示离开 k 个人，每个人会带走平均存款；COLLECT表示向每个人收 k 元，PAY表示支付 k 元。要求任意时刻旅行队人数大于0，且在IN和OUT时，钱数是人数的倍数。输出可能的人数。 $n \leq 50, k \leq 2000$ 。

算法讨论

可以先根据IN和OUT求出人数的下界。在第一次人数改变前和最后一次人数改变后的PAY是没有限制的。在两次人数改变之间的PAY可以看成是同一次PAY。枚举所有可能的PAY进行判断即可。

时空复杂度时间： $O(n\sqrt{k})$ 空间： $O(n + \sqrt{k})$

ACM-ICPC World Finals 2007

试题编号

2007 A

题目名称

Consanguine Calculations

题目大意

每次给出父母、孩子一共三个人中两个人的血型，判断另一个人可能是什么血型。

算法讨论

预处理出若父亲为血型 S_1 ，母亲为血型 S_2 ，孩子为血型 S_3 是否可能，每次询问暴力枚举一下即可。

时空复杂度

时间： $O(1)$

空间： $O(1)$

试题编号2007 *I***题目名称**

Water Tanks

题目大意

从左到右依次有 n 个水箱，每相邻两个水箱之间用一根管道相连，管道的高度是递增的。第一个水箱上面不封顶，后面的水箱都是封顶的。请问将第一个水箱注满水一共要用多少单位的水。 $n \leq 10$ 。

算法讨论

因为管道高度是递增的，因此可以从左到右确定每个水箱中水的高度。水箱中的水位会先达到右边管道的高度，然后将右边的水箱中的水注到管道高度，之后会在当前水箱继续上升，分三种情况讨论，解方程或二分都可以。要注意管道高度等于水箱高度的情况。

时空复杂度时间: $O(n \log \frac{1}{\epsilon ps})$ 空间: $O(n)$

试题编号2007 *J***题目名称**

Tunnels

题目大意

一个间谍现在想逃出你的基地。在间谍移动的过程中，你可以炸毁一些边把间谍困住，目的是不让他移动到0号点。你的基地有 $n + 1$ 个点， m 条边， $1 \leq n \leq 50$, $1 \leq m \leq 2000$ 。

算法讨论

用 $f[i]$ 表示现在间谍在 i 号点，最少需要炸毁几条道路。我们可以做出这样的策略：间谍在 k 号点，先炸毁一些边，然后当间谍走到 i 号点时，炸毁 i 号点与0号点最小割上的边。我们可以假设剩下的点的 f 值都小于等于 x ，那么第一步需要炸毁的边数就是去掉这些点后， k 号点到0号点的最小割。开始时可以对每个点求一遍到0号点的最小割，作为 f 的初始值。注意到若 $f[k]$ 是当前 f 集合的最小值，那么 f 集合中的其他数必然不能更新 $f[k]$ ，于是可以类似于dijkstra算法，求出 $f[1]$ 。

时空复杂度时间: $O(n^4m)$ 空间: $O(n + m)$

ACM-ICPC World Finals 2008

试题编号

2008 A

题目名称

Air Conditioning Machinery

题目大意

一个长方体空间，三维坐标 X, Y, Z 都小于等于20。有一个入口，一个出口。你现有6个L字型（每个有4个单位的立方体）的管道，要从起点连接到终点，问最少要用多少个。

算法讨论

在确定当前一个面之后，下一个L型管道有8种放法，直接搜索即可。

时空复杂度

时间： $O(8^6)$

空间： $O(XYZ)$

试题编号

2008 B

题目名称

Always an Integer

题目大意

给定一个自变量为 n 的 d 次整数系数多项式，判断这个多项式在 n 取任意正整数时，是否恒为 P 的倍数。 $1 \leq d \leq 100$ 。

算法讨论

解决此题需要用到如下命题:一个 d 次多项式 $f(x)$ 是否恒关于 P 与0同余的充要条件是 $f(0) \dots f(d)$ 关于 P 与0同余。证明如下:

必要性显然。我们用归纳法证明充分性。对于一个0次多项式 $f(x)$ ，若 $f(0) \equiv 0 \pmod{P}$ ，那么显然 $f(x) \equiv 0 \pmod{P}$ ，因为 $f(x)$ 的值与 x 无关。

对于一个 d 次多项式 $f(x)$ ，若

$$f(k) \equiv 0 \pmod{P} \quad (0 \leq k \leq d)$$

则

$$f(k+1) - f(k) \equiv 0 \pmod{P} \quad (0 \leq k < d)$$

由于 $f(x+1) - f(x)$ 是一个 $d-1$ 次多项式，那么

$$f(x+1) - f(x) \equiv 0 \pmod{P}$$

又有 $f(0) \equiv 0 \pmod{P}$ ，所以 $f(x) \equiv 0 \pmod{P}$ 。证毕。

时空复杂度

时间: $O(d^2)$

空间: $O(d)$

试题编号2008 *D***题目名称**

The Hare and the Hounds

题目大意

一张图，要采用如下规则走：如果一个点不是选择点，那么选择一条改变方向最少的道路走；如果是选择点，那么按照上面的规则走，直到碰到一个确认点；如果碰到另外一个选择点，或走的距离超过 $maxdis$ ，或遇到死胡同时，要反回本来那个选择点。请你输出要走的长度和正确道路的长度。点数 n 、边数 $m \leq 150$ ， $maxdis \leq 2000$ 。

算法讨论

用一个类似栈的结构存下路径和选择点。按照题目意思模拟即可。

时空复杂度

时间： $O(n \cdot m \cdot maxdis)$

空间： $O(n + m + maxdis)$

试题编号2008 *F***题目名称**

Glenbow Museum

题目大意

用一个 OR 序列可以表示一组多边形， O 表示内角为270度， R 表示内角为90度。我们认为一个序列是合法的，当且仅当存在一个对应的多边形，使得从这个多边形内部的某个点可以看到整个多边形的内部。现给出一个长度 L ，表示序列的长度，问有多少个合法的序列。 $1 \leq L \leq 1000$ 。

算法讨论

容易知道，一个序列是合法的，当且仅当 R 的个数是 $\frac{n}{2} + 2$ ， O 的个数是 $\frac{n}{2} - 2$ ，且没有任何两个 O 相邻(包括开头的和结尾的)。可以看作先放了 $\frac{n}{2} + 2$ 个 R ，然后在空位插入 $\frac{n}{2} - 2$ 个 O 。注意还要减去开头和结尾都是 O 的情况。所以答案为 $C_{\frac{n}{2}+3}^{\frac{n}{2}-2} - C_{\frac{n}{2}+1}^{\frac{n}{2}-4}$

时空复杂度时间: $O(l)$ 空间: $O(1)$

试题编号

2008 G

题目名称

Net Loss

题目大意

给定一个函数 $f(x)$ ，以及一个横坐标 c ，求出两个一次函数 $g_1(x)$ 与 $g_2(x)$ ，使它们交点的横坐标为 c ，且

$$\int_{-1}^c (f(x) - g_1(x))^2 + \int_c^1 (f(x) - g_2(x))^2$$

最小。保证 $f(x)$ 的次数小于等于10，且系数的绝对值小于等于1。

算法讨论

首先有一个很暴力的想法，枚举这两个一次函数经过的点 $(-1, y_1)$, (c, y_2) , $(1, y_3)$ ，再计算代价。计算代价要用到积分，我们可以用 y_1, y_2, y_3 代到积分式中，发现 y_1, y_2, y_3 的最高次幂都为2，所以最终答案关于三个变量都是单峰的。我们可以三分 y_2 ，再分别求 y_1, y_3 的极值点。

时空复杂度

时间： $O(n \log^2 \frac{1}{\epsilon})$

空间： $O(n)$

试题编号2008 *H***题目名称**

Painter

题目大意

有 n 个三角形，要求判断这些三角形是否相交，若不相交，求一个三角形最多被几个三角形所包含。 $n \leq 100000$ 。

算法讨论

判断一些线段是否相交是有 $n \log n$ 的经典做法的。以线段的端点作为关键点，扫描到左端点时将线段加入集合，扫描到右端点时将线段从集合删除，集合内部用线段的 y 坐标为序建立平衡树。每当线段加入集合时，判断一下它与它上面的线段和下面的线段是否有交点；当线段从集合删除时，判断它上面的线段与下面的线段是否有交点。

对于第二个子问题，可以同样用扫描线维护括号序列。如果当前插入的线段是左括号，下面的线段也是左括号，那么这两条线段是兄弟关系；反之，是父子关系，那么就可以直接求出深度了。

时空复杂度时间： $O(n \log n)$ 空间： $O(n)$

试题编号

2008 *J*

题目名称

The Sky is the Limit

题目大意

给定 n 个山脉，求出从上往下看看到的山脉轮廓线的长度。 $1 \leq n \leq 100$

算法讨论

这题有很多做法，我选择了比较暴力但是比较方便的方法。我们把所有山脉的顶点以及交点求出来作为关键点。两个关键点之间的山脉必然互不相交，因此可以很方便求出一个最高的，求出在这两点之间的长度，累加进答案。

时空复杂度

时间： $O(n^3)$

空间： $O(n^2)$

ACM-ICPC World Finals 2009

试题编号2009 *G***题目名称**

House of Cards

题目大意

Axel和Birgit两个人在进行一个游戏，规则是这样的：轮流从牌顶拿走一张牌，可以留着这张牌，可以在两个上三角之间搭一张牌，也可以在一张横向的牌上搭一个上三角形。每次构成新的三角形的时候，如果这个三角形中是黑色的牌多，那么Birgit获得三张牌总和的分数；反之Axel获得三张牌总和的分数。如果牌堆没牌的时候两人手上还有牌，那么Birgit获得黑色牌的总和分数，Axel获得红色牌总和的分数。一开始已经有8张牌组成了4个上三角，加上这8张牌最多只有26张牌，分别是两种颜色的1 K。求两人在最优策略下，Axel的得分。保证牌数 $2m$ 是偶数。 $5 \leq m \leq 13$ 。

算法讨论

这是一道博弈题，数据范围比较小，可以采用带 $AlphaBeta$ 剪枝的最大-最小搜索来做。

时空复杂度时间: $O(2^{2m})$ 空间: $O(m)$

试题编号2009 *H***题目名称**

The Ministers' Major Mess

题目大意

有 m 个大臣， n 个决议，每个大臣会对 k 个决议表决同意或者拒绝。要求每个大臣都有大于一半的决议如他自己所愿，问每个决议是否能通过。 $1 \leq n \leq 100$, $1 \leq m \leq 500$, $1 \leq k \leq 4$ 。

算法讨论

当 $k \leq 2$ 时，这个大臣所有决议都要被满足；否则，最多只有一个决议不被满足，也就是这些决议两两之间都至少一个被满足。可以用2-SAT解决。

时空复杂度时间: $O(n(n + m))$ 空间: $O(n + m)$

试题编号2009 *I***题目名称**

Struts and Springs

题目大意

有 n 个矩形，它们只有包含与分离的关系，并且最外面一个矩形包含所有矩形。每个矩形与最小的包含它的矩形由木板或弹簧连接，矩形内部相对的两边之间也会由木板或弹簧连接。同一方向上，弹簧的压缩比例相等。矩形由外向内变化。现在 m 次改变最外面一个矩形的形状，问你每个矩形的位置和形状。 $n, m \leq 500$ 。

算法讨论

注意到每个矩形新的位置只与原来各条支架长度的比值、当前外围的矩形的位置和大小有关。因此我们只需先预处理出每个矩形外围的矩形是哪一个，每次从大的矩形到小的矩形求出每个矩形的位置和大小即可。需要注意的是， x 轴正方向是右边， y 轴正方向是下边。

时空复杂度时间： $O(n(n + m))$ 空间： $O(n)$

ACM-ICPC World Finals 2010

试题编号2010 *B***题目名称**

Barcodes

题目大意

给定一个被 $Code - 11$ 编码过的条形码，判断它的合法性或输出它原来的值。编码的规则大概是这样，数字0-9以及 $-$ 与开始结束标志分别对应5位01串，每两个01串之间用一个0分开。0表示窄，1表示宽，窄的宽度是宽的一半，宽度可能会有5%的误差（这里的宽窄表示条形码的宽窄）。长度 $n \leq 150$ 。

算法讨论

先把宽度转换为01串（也是最关键的一步）。可以知道，离最大宽度近的必然是1，离最小宽度近的必然是0。令标准窄宽度为 x ，那么标准宽宽度为 $2x$ ，可以解关于 x 的不等式，判断 x 是否存在。接下去的工作就比较容易了，注意细节即可。

时空复杂度时间： $O(n)$ 空间： $O(n)$

试题编号2010 *D***题目名称**

Castles

题目大意

有 n 个城堡，形成一个树结构。军队可以从任意位置出发，按照 dfs 顺序遍历；到达一个城堡时需要 x 个士兵，占领后会减少 y 个士兵，问最少需要多少士兵才能占领所有城堡。 $1 \leq n \leq 100$ 。

算法讨论

每个节点有两个属性：需要的士兵数 x 和减少的士兵数 y 。用 $fx[i]$ 表示占领完 i 的子树需要的士兵数， $fy[i]$ 表示占领完 i 的子树会减少的士兵数。如果我们依次占领 i 的子树，会发现按照子树 $fx[k] - fy[k]$ 的逆序行军是最优的，证明很简单，可以用调整法证明。

时空复杂度时间： $O(n^2 \log n)$ 空间： $O(n)$

试题编号2010 *F***题目名称**

Contour Mapping

题目大意

地图上有一些测高点，共 s 行，奇数行有 p 个，偶数行有 $p + 1$ 个，相邻两行之间的测高点形成一些正三角形的平面区域，正三角形的边长为 d 。你要给这些区域画一些等高线，只要是 h 的倍数都要画出来。如果一块区域都是同一高度，那么只需要在区域的边界画等高线。 $1 \leq s \leq 100$, $1 \leq p \leq 100$, $1 \leq d \leq 10$, $1 \leq h \leq 1000$, 所有测高点的高度 ≤ 1000000 。

算法讨论

我们可以依次考虑每个三角形。如果三个测高点高度都不同，那么可以计算出三角形内部顶点在某两条边上的最短的等高线长度、最长的等高线长度，以及等高线条数。由于剩下的等高线是这个梯形腰上距离相等的点连成的，因此是等差的，可以直接计算总和。如果三角形有两个测高点相等且是 h 的倍数，那么要考虑旁边那个三角形的形态。如果不存在旁边那个三角形或者旁边那个三角形三个测高点都相等，那么应累加 d 的代价，否则是 $\frac{d}{2}$ 的代价。如果一个三角形三个测高点都相等且都是 h 的倍数，那么只有不存在旁边的三角形会累加 d 的代价。还有一点值得注意的是，如果一个三角形第二小的测高点是 h 的倍数，那么会存在两边最长的等高线重合的情况，应该把代价去掉。

时空复杂度时间: $O(sp)$ 空间: $O(sp)$

试题编号2010 *H***题目名称**

Rain

题目大意

一张 n 个节点的平面图，每个区域都是三角形。每个节点有一个高度 h 。现在下了很大的雨（水量无限），问组成了海拔为多少的湖，输出每一个湖。我们认为这个平面区域外的地方高度都是无穷小。 $n \leq 2704$ 。

算法讨论

一个点高度能为 h 的充要条件为：1.这个点的高度小于 h ；2.不存在一条通向边界的路，使这条路上的所有点高度都小于 h 。两个点在一个湖内的充要条件为：存在一条高度都小于 h 的路径连接这两个点。那么我们从边界开始，做类似与最短路的 dp ，求出每个点可能的最大高度，然后再将在同一个湖里的点并起来就好了。

时空复杂度时间： $O(n^2)$ 空间： $O(n)$

试题编号2010 *J***题目名称**

Sharing Chocolate

题目大意

你有一块大小为 $x * y$ 的巧克力，要将它分成 n 个小块，告诉你每一块的面积，问是否可能。 $1 \leq x, y \leq 100, 1 \leq n \leq 15$ 。

算法讨论

利用 dp 的思想，令 $dp[S][i]$ 表示当前用了在 S 集合内的小巧克力，拼成一边长为 i 的巧克力是否可能。由于这个是布尔类型，可以将所有可能的 i 压成一个大数字。转移时枚举子集，若两个子集答案 x 都可以，那么这个集合答案为 x 也是可行的。枚举完子集后再将判断是否可以将矩形转向就可以了。

时空复杂度时间： $O(3^n)$ 空间： $O(2^n)$

试题编号2010 *K***题目名称**

Paperweight

题目大意

一个由两个四面体构成的多面体，内部有一个芯片，你要求出它离桌面的最大值和最小值。注意一种放置方法是稳定的，当且仅当中心往任意方向移动0.2个单位都是不会移动的。坐标的绝对值 ≤ 1000 。

算法讨论

枚举每三个点作为底面，判断另外两个点是否在这个面同侧。接下去求出重心到底面的投影，判断投影是否在底面的内部，且到底面的边的最小距离是否大于0.2。如果这种放置方法可行，那么用当前 F 点到底面的距离更新答案。需要注意的是，底面可能是四边形，更特别的，也有可能是凹四边形。

时空复杂度时间: $O(1)$ 空间: $O(1)$

ACM-ICPC World Finals 2011

试题编号

2011 A

题目名称

To Add or to Multiply

题目大意

有一个程序，只有两种字符： A 、 M 。 A 表示将当前数字加上 a ， M 表示将当前数字乘上 m 。保证输入是 p 到 q 之间的整数，要你保证输出是 r 到 s 之间的整数。如果有多组答案，输出最短的那个；如果还有多组，输出字典序最小的。 $1 \leq a, m, p, q, r, s \leq 1000000000$ 。

算法讨论

由于无论什么时刻，数字肯定是正数，因此经过 A 和 M 的操作，大小关系是不会变的。因此只要使 p 、 q 都能满足即可。如果初始数字是 x ，那么经过变换后肯定变为 $x * m^? + \sum(?am^?)(?$ 表示整数)。因此可以先枚举进行了几次乘操作，那么可以求出 $\sum(?am^?)$ 的范围，将其变为 $\sum(?m^?)$ 的范围，转换成 m 进制后，就是 m 进制的数字个数。对于上界 r 和下界 l ，找出它们第一个不相等位 k 。如果从 $k+1$ 位开始 l 都是0，那么 l 就是最优的；否则，将第 k 位加1，从 $k+1$ 位开始都变为0就是最优的。

时空复杂度时间： $O(\log^2 s)$ 空间： $O(\log s)$

试题编号2011 *E***题目名称**

Coffee Central

题目大意

一个 $dx * dy$ 的街道，其中有 n 个咖啡馆。有 q 个人，每个人有一个最长走动的曼哈顿距离 m 。请在 $dx * dy$ 个十字路口中选择一个位置，使他能走到的咖啡馆最多。 $1 \leq dx, dy \leq 1000, 1 \leq n \leq 500000, 1 \leq q \leq 20$ 。

算法讨论

对于每次询问，我们可以求出 $f(x, y)$ ，表示站在 (x, y) 可以走到的咖啡馆数。在我们求出 $f(x-1, y)$ 后，只要加减一些斜条就能得到 $f(x, y)$ ，当然也可以由 $f(x, y-1)$ 推出 $f(x, y)$ 。

时空复杂度时间: $O(q \cdot dx \cdot dy)$ 空间: $O(dx \cdot dy)$

试题编号2011 *H***题目名称**

Mining Your Own Business

题目大意

你有一个矿场，有 m 条矿道，连接 n 个挖矿点。现在你要对其中一些挖矿点建造竖井，使得任何一个挖矿点塌了，在其他挖矿点的工人都可以走到竖井。请你输出最少需要建造多少个安全竖井，并输出方案数。 $m \leq 50000$ 。

算法讨论

先求出原图中所有割点。如果原图没有割点，那么任选两个挖矿点都可以。否则，这些割点将原图分成了树结构，最少需要建造叶子个数个竖井，方案数为叶子中节点个数的积。

时空复杂度时间： $O(m)$ 空间： $O(m)$

试题编号

2011 I

题目名称

Mummy Madness

题目大意

你现在在坐标原点，四周有 n 个木乃伊。每一次都是你先选择一个位置走或者不走，然后每个木乃伊走向离你最接近的位置。你会尽可能推迟木乃伊抓到你。这个图是八连通的，判断木乃伊要多久才能抓到你。 $1 \leq n \leq 100000$ 。

算法讨论

观察到如果你不动，一个木乃伊走到你的时间是 $\max(dx, dy)$ ，那么其实 x 和 y 是互不干扰的。我们可以先二分时间。在某一时刻木乃伊没抓到你当且仅当所有木乃伊在 x 轴上没抓到你或在 y 轴上没抓到你。可以先枚举 x 的值，再用堆维护 y 的值的范围，判断有无解即可。

时空复杂度时间： $O(n \log^2 n)$ 空间： $O(n)$

ACM-ICPC World Finals 2012

试题编号

2012 A

题目名称

Asteroid Rangers

题目大意

空间上有 n 个点，每个点有一个初始位置和一个移动向量，问这个图的最小生成树改变了几次。 $n \leq 50$ 。

算法讨论

只有两条边的大小改变的时候最小生成树才会变化，这样的改变次数为 n^4 次。我们可以在每次改变的时候记下这棵树的 dfs 序，那么改变能否影响树的结构可以在 $O(1)$ 的时间内得到。事实证明树改变的次数不会很多。

时空复杂度时间: $O(n^4 \log n)$ 空间: $O(n^4)$

试题编号

2012 C

题目名称

Bus Tour

题目大意

一张 n 个点的图，第2个到第 $n-1$ 个是旅馆，第一个是起点，最后一个为终点，车要从起点到终点再到起点。要求最先上车的 $\lfloor \frac{n-2}{2} \rfloor$ 个人也是最先下车的 $\lfloor \frac{n-2}{2} \rfloor$ 个人，车可以经过旅馆但是不停下。求最短路。 $3 \leq n \leq 20$ 。

算法讨论

可以先用 $floyd$ 求出任意两点间的最短路。用 $dp[S][i]$ 表示从起点出发，已经经过了 S 集合的旅馆，当前在 i 号点的最短路径。 $dp1[S][i]$ 表示从终点出发，已经经过了 S 集合的旅馆，当前在 i 号点的最短路径。之后只要枚举最先上车的 $\lfloor \frac{n-2}{2} \rfloor$ 个人就可以更新答案了。

时空复杂度时间： $O(n2^n)$ 空间： $O(n2^n)$

试题编号

2012 *E*

题目名称

Infiltration

题目大意

有 n 个单位，任意两个单位 A, B ，要么 A 能控制 B ，要么 B 能控制 A ，当然 A 也能控制自己，求最少要选择多少个单位，才能控制所有单位（直接控制）。 $n \leq 75$ 。

算法讨论

一共有 $\frac{n(n+1)}{2}$ 对控制关系，那么至少有一个单位能控制 $\lceil \frac{n+1}{2} \rceil$ 及以上个单位。如果每次去掉这个单位，那么最多剩下 $\lfloor \frac{n-1}{2} \rfloor$ 个单位。从75开始，会变成37、18、8、3、1、0。所以答案小于等于6。我们枚举小于等于5个单位，如果能找到一组解，那么最优解小于等于5。否则按上面的方法构造一个答案等于6的解。注意搜索时要压位。

时空复杂度

时间： $O(C_n^5)$ 空间： $O(n)$

试题编号

2012 L

题目名称

Takeover Wars

题目大意

有两个集团 X 和 Y ， X 旗下有 n 个子公司， Y 旗下有 m 个子公司，每个子公司有一个市场价值。这两个集团正在进行商业博弈，每一轮由 X 集团先进行操作，再由 Y 集团进行操作。操作分为两种：1.合并操作：将自己旗下的两个子公司合并，变成一个新的子公司，新的子公司的市场价值为两个旧子公司的和；2.吞并操作：可以将对方的一个市场价值比自己小的子公司吞并，吞并后对方的子公司消失，自己的市场价值不变。当然在某一刻也可以选择不进行操作。当某个集团没有子公司时，这个集团就输了。问 X 集团能否取得胜利。 $1 \leq n, m \leq 100000$ 。

算法讨论

注意到每个集团的最终目的就是培养一个足够优秀的子公司，来吃掉对方所有的公司。因此可以得出如下贪心策略：1.若要进行合并操作，肯定是合并市场价值最高的两个子公司，因为是否能赢只与市场价值最高的子公司有关；2.若进行吞并操作，肯定是吞并对方市场价值最高的子公司，因为对方需要的也正是这个子公司。

光有这两个结论还不够。又注意到只要吞并对方一个合并而成的子公司，就能稳操胜券。那么若两者有一方进行过合并操作后，两者的策略就都为一直合并，直到能吞并对方市场价值最高的公司。在进行第一步操作之后，肯定有一个公司进行了合并操作（否则 X 的最大值大于 Y 的最大值且 Y 的次大值大于 X 的最大值，矛盾），这样两个公司都会按照上述的策略进行操作。

总的算法流程就是先枚举两者第一步是进行吞并操作还是合并操作，之后如果能吞并对方则取得胜利，否则进行合并操作。

时空复杂度时间： $O(n \log n)$ 空间： $O(n)$

ACM-ICPC World Finals 2013

试题编号

2013 A

题目名称

Self-Assembly

题目大意

有 n 个正方体分子，每条边上都有一个属性，要不是大写字母 '+' 或 '-', 要么是 '00'。对于相同的大写字母，如果符号相反，那么这两条边就可以拼起来。问这些分子拼成的大分子是否可能无限。 $n \leq 50000$

算法讨论

首先要知道的是，这些分子是不会叠起来的。因为可以随便转换方向，只要每次都向右、向下堆叠就不会重叠了。因此只要有一条从 S 到达 S 的路径，就可以说这个图无限大。这样考虑有点复杂，不妨将边作为点，那么分子就变成了转移，点数最多是52。暴力判断有没有走向自己的路径即可。

时空复杂度时间: $O(n)$ 空间: $O(1)$

试题编号

2013 B

题目名称

Hey, Better Bettor

题目大意

有一个赌场给出如下的优惠政策：亏损 k 元后可以申请补偿亏损的 $x\%$ ，但只能使用一次，注意这个 k 是指总支出-总收益，且大于0。每一场赌局开始将付1块钱，如果赌赢了将得到2块钱。现在给出每一场赢的概率 $p\%$ ，问你在最优策略下期望能赚多少钱。

算法讨论

稍加分析可知只有在亏损 a 元或赢得 b 元后才会停止赌局。令 $f(x)$ 为在当前已经赢得 x 元的情况下的最大期望收益，容易得到

$$f(-a) = -a(1 - x\%), \quad f(b) = b$$

又有

$$f(x) = p\%f(x+1) + (1 - p\%)f(x-1)$$

移项得

$$f(x) = \frac{f(x-1)}{p-1} - \frac{f(x-2)(1-p\%)}{p\%}$$

令 $g(x) = f(x+a)$ 可以列出特征根方程

$$x^2 = \frac{x}{p\%} - \frac{1-p\%}{p\%}$$

解得 $x_1 = \frac{1-p\%}{p\%}$, $x_2 = 1$, 所以

$$g(x) = c_1 x_1^x + c_2 x_2^x$$

将 $g(0) = -a(1 - x\%)$, $g(a+b) = b$ 代入得

$$\begin{cases} c_1 + c_2 = -a(1 - x\%) \\ c_1 x_1^{a+b} + c_2 = b \end{cases}$$

解得

$$\begin{cases} c_1 = \frac{b + a(1 - x\%)}{\left(\frac{1-p\%}{p\%}\right)^{a+b} - 1} \\ c_2 = -a(1 - x\%) - \frac{b + a(1 - x\%)}{\left(\frac{1-p\%}{p\%}\right)^{a+b} - 1} \end{cases}$$

于是

$$g(a) = \frac{(b + a(1 - x\%))(\frac{1-p\%}{p\%})^a}{(\frac{1-p\%}{p\%})^{a+b} - 1} - a(1 - x\%)$$

求导可得 $g(a)$ 关于 a, b 都是单峰的，因此可以用三分套三分来解决。

时空复杂度

时间: $O(\log^2 \frac{1}{\epsilon_{ps}})$

空间: $O(1)$

试题编号2013 *D***题目名称**

Factors

题目大意

令 $f(x)$ 表示将 x 分解质因数后，这些质因数的排列个数。输入 $k = f(x)$ ，求最小的 x 。 $1 \leq k, x \leq 2^{63}$

算法讨论

令 $x = \prod p_i^{a_i}$ ，那么 $f(x) = \prod \binom{sum_{k \leq i} a_k}{a_i}$ ，那么只要枚举每个质因子出现了几次即可，跑得还是挺快的。

时空复杂度时间： $O(1)$ 空间： $O(1)$

试题编号2013 *E***题目名称**

Harvard

题目大意

你有两种访问内存的操作。第一个是访问第一个内存库里的单位，第二个是访问 *BSR* 内存库里的单位。还有一个操作是指定 *BSR* 为第 k 个内存库。现在有 b 个内存库，每个内存库最多有 s 个单位。你写了一个程序，有两种操作，共 x 个操作：1. 访问一个内存变量 V_i ；2. 将一段操作循环 n 次。请你输出最少需要的指令数。 $1 \leq b, s \leq 13$, $1 \leq i \leq \min(b * s, 13)$, $1 \leq n \leq 1000000$, $x \leq 1000$ 。

算法讨论

可以先枚举哪些变量属于第一个内存库，就可以处理出 i 变量与 j 变量相邻的有几组。之后再暴力枚举每个变量属于哪个内存库即可。

时空复杂度时间: $O(x \cdot \max i^4 + \max i^{13})$ 空间: $O(x \cdot \max i^2)$

试题编号2013 *F***题目名称**

Low Power

题目大意

n 个机器，每个机器有两个芯片，每个芯片由 k 个电池组成。每个芯片的能量是 k 个电池能量的最小值。求所有机器两个芯片能量之差的最大值的最小值。 $2nk \leq 10^6, 1 \leq p_i \leq 10^9$

算法讨论

只要考虑每个芯片能量最小的电池即可。一个机器的两个芯片最小电池能量一定是相邻的，而且越靠前越好，因为这样作为最小值的机会就变大了。可以二分答案，之后可以得到哪些电池可以作为芯片的最小值，再判断这些电池作为最小值的情况是否可能。

时空复杂度时间: $O(n \log n)$ 空间: $O(n)$

试题编号2013 *H***题目名称**

Matryoshka

题目大意

有 n 个玩偶，每个玩偶有一个大小。每次可以合并相邻两个玩偶，合并的代价为需要拆开的玩偶的个数。只有当一个玩偶中的玩偶大小是 $1 \dots m$ 时，才算是一个合法的套娃。问将这些玩偶组成一些套娃的最小代价。 $n \leq 500$

算法讨论

这是个经典的 dp 问题，只要能把两个套娃合并的代价求出来，就能在 n^3 的时间内完成 dp 。容易发现，除了最小的一些套娃不用拆开外，别的都需要拆开。因此代价就是两边总的玩偶数减去两边分别比另外一边的最小值还小的个数。当然还得预处理判断每个区间是否可行，时间复杂度不会变。

时空复杂度时间: $O(n^3)$ 空间: $O(n^2)$

试题编号

2013 *J*

题目名称

Pollution Solution

题目大意

求一个 n 边形和一个半圆的交的面积。 $n \leq 100$ 。

算法讨论

可以直接用自适应辛普森做。

时空复杂度

时间: $O(n \log \frac{1}{\epsilon})$

空间: $O(n)$

试题编号2013 *K***题目名称**

Up a Tree

题目大意

有一个人想对一棵二叉树实现输出先序、中序、后序，但是他把递归函数搞乱了（就是先序里递归中序之类）。现在给出他的程序跑出的先序、中序、后序遍历，用大写字母表示，请你输出所有可能的写法，以及这种写法中字典序最小的树的先序、中序、后序。树的字典序定义为将先序、中序、后序依次拼起来后的字典序。

算法讨论

首先应该枚举他的程序是怎样写的。知道是怎样写的之后， $dp[S1][S2][S3]$ 表示他所谓的先序是 $S1$ ，中序是 $S2$ ，后序是 $S3$ ，这样是否可能，以及如果可能，字典序最小的树的先序，中序，后序是多少。由于状态数很多，应采用记忆化搜索。

时空复杂度时间： $O(26^8)$ 空间： $O(26^6)$