*IOI*2014 中国国家集训队第一次作业

第二部分 泛做表格

Accepted(159/159)

绍兴一中　董宏华

2014 年 1 月 25 日

Contents

Chapter 1

ACM/ICPC World Finals 2013 (11/11)

1.1 [A]Self-Assembly

1.1.1 题意简述

正方形四面贴形如“符号+字符”（符号为 $+-$，字符为大写 $A \sim Z$）的标签，只有符号相反，字符相同的两个标签所在边才能贴在一起。但特殊标签“00”和所有标签都不能贴在一起。问用无限个给出 n 种的正方形，能否拼成无限的图案。

【数据范围】$n \leq 40000$。

1.1.2 算法分析

如果可以构成循环，那么图案就是无限的。如果把正方形中的一个标签到其他标签的配对标签看作一条边，那么就相当于找一条从某个标签回到它自己的路径，可以证明只要有路径一定可以一直往右上方向连构成循环，而不会回到它本身。所以只要在读入每种正方形后连边，用 $floyd$ 判断能否从自己走到自己即可。

【时间复杂度】$O((2 \times |S|)^3 + 4^2 n)$，【空间复杂度】$O((2 \times |S|)^2)$。

1.2 [B]Hey, Better Bettor

1.2.1 题意简述

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。

【数据范围】$0 \le x < 100, 0 \le p < 50$。

1.2.2 算法分析

由于当前状态只和钱有关，可知只有在亏损 a 元或赢 b 元之后才会停止赌博。首先若 p 为 0 ，则答案为 0 ，下面不考虑 $p = 0$ 的情况。令 p 为读入的 $p/100$, $\lambda = 1 - x\%$, $f(x)$ 为在当前赢得 x 元的情况下的最大期望收益，则 $f(-a) = -\lambda a, f(b) = b$ 。

又有

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

移项得

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

令 $g(x) = f(x-a)$ ，列出特征根方程

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

解得 $x_1 = 1, x_2 = \frac{1-p}{p}$, $\because p < 0.5$, $\therefore x_1 \ne x_2$

$$\therefore g(n) = c_1 {x_1}^n + c_2 {x_2}^n = c_1 + c_2 \left(\frac{1-p}{p}\right)^n$$

将 $g(0) = f(-a) = -\lambda a, g(a+b) = f(b) = b$ 代入得

$$c_1 + c_2 = -\lambda a, c_1 + c_2 \left(\frac{1-p}{p}\right)^{a+b} = b \text{ , } \therefore c_2 = \frac{b + \lambda a}{\left(\frac{1-p}{p}\right)^{a+b} - 1}$$

$$\therefore f(0) = g(a) = c_1 + c_2\left(\frac{1-p}{p}\right)^a = \frac{(b+\lambda a)\left(\left(\frac{1-p}{p}\right)^a - 1\right)}{\left(\frac{1-p}{p}\right)^{a+b} - 1} - \lambda a$$

由打表或者求导可知 $f(0)$ 关于 a, b 都是单峰函数，又由于 a, b 均为整数，可用二分套二分。

关于二分上界，由偏导为 0 可大概得出为 $\frac{10}{\log_{10}\left(\frac{1-p}{p}\right)}$。

【时间复杂度】$O(\log^2)$，【空间复杂度】$O(1)$。

1.3 [C]Surely You Congest

1.3.1 题意简述

n 个顶点，m 条边的带权无向图，有 c 辆车要分别从某些顶点以他所能达到的最短时间（无视其他车的干扰时）到达 1 号点。不能有两辆车在同一时间从同一顶点沿同一条边行驶。若所有车同时出发，问最多有多少车可以按时到达 1 号点。

【数据范围】$n \leq 25000$, $m \leq 50000$, $c \leq 1000$, $1 \leq$ 边权 ≤ 10000。

1.3.2 算法分析

令 d_i 为顶点 i 到顶点 1 的最短路，取出满足 $d_i + w_{i,j} = d_j$ 的边 (i, j) ，在新建的图中加入有向边 (j, i) 。容易发现新建的图为 DAG ，并且沿着图中的有向边走到顶点 1 肯定是最短路径。

然后考虑开始时在顶点 i 和顶点 j 的两个人，假如他们同时到达了顶点 k ，容易说明 $d_i = d_j$ 。并且如果 $d_i = d_j$ ，他们到达相同顶点时的时间也一定相同。这样的话，将人按照出发顶点的 d 值分组，不同组的人是不会影响的。每组的最优解是个简单的最大流问题。

由于时限 $10s$ ，直接暴力做最大流就能 AC 了。

【时间复杂度】$O(spfa(n, m) + c \times MaxFlow(DAG(n, m)))$，

【空间复杂度】$O(n + m)$。

1.4 [D]Factors

1.4.1 题意简述

定义 $f(x)$ 为把 x 分解成质数相乘的形式不同的排列方案数，如 $20 = 2\times2\times5 = 2\times5\times2 = 5\times2\times2$，则 $f(20) = 3$。给出 T 组 n，求最小的 x 满足 $f(x) = n$。

【数据范围】$T \le 1000$, $n, x \le 2^{63}$。

1.4.2 算法分析

若

$$x = \prod_{i=1}^{k} {p_i}^{q_i}$$

则

$$f(x) = \frac{(\sum_{i=1}^{k} q_i)!}{\prod_{i=1}^{k} q_i!}$$

因为要满足 x 最小，所以 q_i 不递增，又因为 $x \le 2^{63}$，所以实际有效的 x 不多，直接 dfs 预处理出所有有效的 x，按 $f(x)$ 存入 $Hash$ 表中，保留最小的 x。询问直接查询 $Hash$ 表即可。可用 STL 的 map 替代 $Hash$ 表。

【时间复杂度】$O(\ (|S| + T)[\log|S|]\ )$，【空间复杂度】$O(|S'|)$。

其中 $|S|$ 为有效的 x 的数量，$|S'|$ 为有效的 $f(x)$ 的数量。

1.5 [E]Harvard

1.5.1 题意简述

有 n 个内存块，每个内存块有 m 个内存。在内存中读取变量需要进行操作，有 3 种操作：读取 0 号内存块的某个内存上的变量、读取 x 号内存块的某个内存上的变量、指定 x 的值。给 s 个变量分配内存，使得仅包含循环和读取变量的语句数为 L 的程序需要的操作数最少。初始 x 的值未定义。

【数据范围】$n, m, s \le 13$，$L \le 10^3$，循环展开后语句数 $\le 10^{12}$。

1.5.2 算法分析

首先所有读取变量都需要一次操作，所以只需计算相邻两次需要用到第二个操作但 x 不同的次数。所以操作次数只和一个变量和哪些变量在同一个内存块中有关。而 0 号内存块比较特殊，放入其中的变量的读取操作可以忽略。所以先 dfs 确定 0 号内存块中的变量，然后用栈预先算出程序中变量两两相邻（在去掉在 0 号内存块中的变量后）的次数，最后 dfs 其余的分组方式，边 dfs 边计算需要的操作数，方便进行最优性剪枝。

【时间复杂度】$O(\binom{s}{m} \times (L + bell(s-m) \times s^2))$，【空间复杂度】$O(L)$。

1.6 [F]Low Power

1.6.1 题意简述

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。

【数据范围】$2nk \leq 10^6$, $p_i \leq 10^9$。

1.6.2 算法分析

答案显然满足二分性质，设二分值为 x 。

假设最后选了 $2n$ 个关键电池作为每个芯片的最小值，最优方案中一定是能量相邻的两个配对。

那么从前往后决定每个电池是否为关键电池。在选了某个配对中的第一个电池后，第二个电池的能量要尽可能与第一个电池接近，而关键电池的能量越小，方案越容易合法。所以这时一定是把紧接的一个电池设为关键电池。

那么可以 DP ，用 $f[i]$ 表示前 i 个电池中最多选了几对关键电池，而且要满足 $f[i] \times 2 \times k \geq i$ 才能进行转移。

转移即第 i 个不选，则 $f[i+1] = \max(f[i+1], f[i])$ ，或第 i 个和 $i+1$ 个作为配对的关键电池，且满足 $p[i+1] - p[i] \leq x$ ，则 $f[i+2] = \max(f[i+2], f[i]+1)$ 。

最后判断 $f[2nk] \geq n$ 是否成立即可。

【时间复杂度】$O(n \log n)$，【空间复杂度】$O(n)$。

1.7 [G]Map Tiles

1.7.1 题意简述

把n条边的多边形铺在平面上（不可旋转），使其占用网格数最少。每个网格大小为 $xs \times ys$ 。

【数据范围】$n \leq 50$, $xs, ys \leq 100$, $0 \leq x_i \leq 10xs$, $0 \leq y_i \leq 10ys$。

1.7.2 算法分析

由数据范围可知每行每列最多使用 $m = 11$ 个网格。

肯定有一种最优方案是某些点或边卡在网格的边界上，则要分以下情况讨论：

1. 卡两点，则分别枚举 x, y 方向卡住的两点后计算。

2. 卡点边，需要枚举 x 或 y 方向卡住的点，然后枚举边，并枚举在边上卡住的点相对于点在卡住方向上的距离（肯定为整数），求交点以确定卡住的点。

3. 卡两边，需要枚举两边，然后枚举卡在边上的两个点在 x, y 方向的距离（肯定为整数），将一边平移后求交点确定卡住的点。

这样总共有 $O(n^2m^2)$ 种卡住的方案，但有较多重复，可将偏移量取小数部分，用 *set* 去重。

现在已经得到了最终地图的位置，需要计算其覆盖的网格个数，分两类：被边穿越的，或在多边形内部的。

其中被穿越可以用网格的 4 条边和多边形的边分别求交判断，要特判刚好卡在边界上的情况。在内部的可以把网格中心作为判定点，用射线法分别与多边形的边求交后判断。这样可以 $O(nm^2)$ 完成判断，加些常数优化以加速。

还有一种 $O(mn\log n)$ 的方法，枚举每行，用多边形每条边和该行顶部和底部求交，记录穿越的交点。把从下到上穿越记为 $+1$，否则记为 -1，把交点排序后可看作括号匹配，括号内部分均为覆盖部分，扫一遍即可求出覆盖网格数。

【时间复杂度】$O(n^3m^4)$　$O(n^3m^3\log n)$，【空间复杂度】$O(n+m^2)$。

1.8 [H] Матрёшка

1.8.1 题意简述

有一行 n 个单层套娃，大小为 a_i，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。

【数据范围】$n, a_i \le 500$。

1.8.2 算法分析

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 $\le j$ 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。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

1.9 [I]Pirate Chest

1.9.1 题意简述

$n \times m$ 的水池，每格深度为 $d_{i,j}$，把底面为 $x \times y$ 的宝箱放入其中（要求 x, y 满足其中一个 $< a$，另一个 $< b$），宝箱会被最浅的格子卡住。若宝箱高度可为任意非负整数，问在保证宝箱不露出水面的情况下，宝箱体积最大为多少。

【数据范围】$a, b, n, m \leq 500$, $d_{i,j} \leq 10^9$。

1.9.2 算法分析

在已知底面，得出宝箱卡住的深度后，可以通过计算直接得出宝箱最大的高度。关键是求卡住的深度。

假设 $a \geq b$，枚举 x，则 $x > b$ 时 y 要满足 $y \leq a$，否则要满足 $y \leq b$。随着 x 的增大，不断合并相邻行，可得到每个 $x \times 1$ 的矩形中的最小值，然后利用这些值每行分别计算。观察式子后发现 x 和卡住深度确定后， y 显然越大越好，所以确定出每个值作为最小值的区间后让 y 尽可能大。而每个值作为最小值的区间可以建成笛卡尔树后得出。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

1.10 [J]Pollution Solution

1.10.1 题意简述

给出一个在 x 轴上方， n 个点的多边形以及一个半径为 r 半圆，求多边形被半圆覆盖的面积。

【数据范围】$n \leq 100, r \leq 1000$。

1.10.2 算法分析

对比多边形面积的求法，我们可以将问题简化为求一个三角形被半圆覆盖的面积，然后根据两个顶点表示的向量的叉积的值，将各个面积加减得到答案。

此时，三角形的一个顶点必在圆心。我们可以分 4 类情况讨论:

1. 三角形其余两个顶点均在半圆内部，那么答案就是三角形的面积；

2. 三角形两个顶点均在半圆外部，并且第三条边与圆弧交点小于两个，那么答案就是一个扇形的面积；

3. 三角形一个顶点在半圆外部，那么答案化为一个三角形和一个扇形的面积和；

4. 三角形两个顶点均在半圆外部，并且第三条边与圆弧有两个交点，那么答案就是一个三角形和两个扇形的面积和。

对于第 2 类和第 4 类的情况可以用点到线段的距离来判定，但是比较麻烦，其实可以将后三类情况一起判断，直接使用解二次方程的方法判断和圆弧交点的个数以及交点的位置，然后第 4 类可以通过垂足拆分转化为第 3 类，第 3 类可通过交点拆分转化为第 2 类，这样计算就比较方便了。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

1.11 [K]Up a Tree

1.11.1 题意简述

给出把树的前序、中序、后序遍历的递归过程中的过程调用打乱（比如前序遍历中调用了对左子树调用了中序遍历过程，而对右子树调用了后序遍历过程）后产生 n 个节点的树的“前序”，“中序”，“后序”遍历。问可能的打乱情况和在这种打乱情况下字典序最小的树的前序、中序、后序遍历（字典序先比前序，再比中序）。

【数据范围】$n \leq 26$。

1.11.2 算法分析

先枚举 6 个递归调用分别是什么，然后用给出的三个输出进行搜索，搜索的参数为错误前序、中序、后序遍历程序对应的输出，由于它的递归调

用可能错的，不能保证到任何一个子树，都存在每种输出，但不难说明至少存在一个输出。这样搜索中可能需要枚举根或者左子树的大小。

用 *map* 进行记忆化。

【时间复杂度】$O(3^3 \times n^4[\times \log n])$，【空间复杂度】$O(3^3 \times n^3)$。

Chapter 2

ACM/ICPC World Finals 2012 (12/12)

2.1 [A]Asteroid Rangers

2.1.1 题意简述

三维坐标上 n 个点，每个点匀速直线运动，求最小生成树改变的次数 $+1$。

【数据范围】$n \leq 50$，$-150 \leq x, y, z \leq 150$，$-100 \leq v_x, v_y, v_z \leq 100$。

2.1.2 算法分析

对于每个事件点，发生大小关系改变的两条边中，定义在事件点之后长度变得更小的边为取代边，另一条为被取代边，如果被取代的边不在当前的最小生成树上，那么最小生成树显然不会发生改变，更进一步地，只有在取代边的两端点在最小生成树的路径中包含被取代边，最小生成树才会发生改变。而一条边树的路径上，相当于这条边断开后，路径两端点不连通，也就是一个在子树内，另一个不在子树内，这样就可以在每次最小生成树改变后维护 dfs 序，$O(1)$ 就可以得到对于边是否在路径上的询问。由于每次最小生成树改变后需要重新得到 dfs 序，每次为 $O(n)$ ，但由于答

案大概不会超过 $O(n^3)$ 的，所以复杂度也为 $O(n^4)$。此外，造出答案超过几百的数据很困难（至少我随机数据得到的答案大概是 $O(n^2)$ 的）。

【时间复杂度】$O(n^4 \log n)$，【空间复杂度】$O(n^4)$。

2.2 [B]Curvy Little Bottles

2.2.1 题意简述

给出一个每一小段都是圆柱形的瓶子的横截面上半部分的形状对应的多项式 $P(x) = \sum_{i=0}^{n} a_i x^i$，以及底部和顶部 x 坐标 x_{low}, x_{high}，求瓶子的体积，并每隔 inc 单位体积标刻度线（最多只标前 8 条），求刻度线相对于底部的 x 坐标。保留两位小数，并保证刻度线之间的距离至少为 0.05。

【数据范围】$n \leq 10$，$-100 \leq a_i \leq 100$，$a_n \neq 0$，$-100 \leq x_{low} < x_{high} \leq 100$，$x_{high} - x_{low} > 0.1$，$1 \leq inc \leq 500$。

2.2.2 算法分析

显然体积= $\int_{x_{low}}^{x_{high}} \pi P(x)^2 \, dx$，把平方暴力展开后计算即可。而标刻度线时要解的是 $2n$ 次方程，直接解有难度，但注意到本题精度要求不高，而且保证了刻度线之间的间距，所以可以按 0.005 暴力枚举，一旦在当前刻度时体积超过 $j \times inc$ 了，则标上第 j 条刻度线。

【时间复杂度】$O(\frac{x_{high} - x_{low}}{eps} \times n^2)$，【空间复杂度】$O(n)$。

2.3 [C]Bus Tour

2.3.1 题意简述

n 个点 m 条边的图中，0 号点是起点，$n-1$ 号点是终点，要从起点遍历其他所有点，然后到达终点，并从终点遍历其他点回到起点，遍历过程中可以中途经过某些点但不算作到达。要求第一次遍历的前 $\lfloor \frac{n-2}{2} \rfloor$ 个点在第二次遍历中也是前 $\lfloor \frac{n-2}{2} \rfloor$ 个，求总路程最短是多少。

【数据范围】$n \leq 20$。

2.3.2 算法分析

这个问题和 TSP 问题很像，而 TSP 问题可以用 $O(2^n \times n^2)$ 的复杂度解决。所以使用类似的方法解决此题。先 $floyd$ 预处理，那么就可以得到两两之间实际的最短距离了（中转点只算途径）。用 $f[i][j]$ 表示从起点出发，遍历过的点的二进制状态为 i，当前在 j 号点，最小的路程和为多少，转移时只需枚举一个还未经过的点作为下一个点即可。同样地 $g[i][j]$ 表示从终点出发的情况。最后，枚举一个大小为 $\lfloor \frac{n-2}{2} \rfloor$ 的子集作为先经过的部分，剩余部分为后经过的部分，分别枚举两部分最后到达的点，找出两次遍历的最小值，相加取 min 即可。

【时间复杂度】$O(2^n \times n^2)$，【空间复杂度】$O(2^{n-1} \times n)$。

2.4 [D]Fibonacci Words

2.4.1 题意简述

$F(n)$ 为 $string$，定义 $F(n) = \begin{cases} 0 & if\ n = 0 \\ 1 & if\ n = 1 \\ F(n-1) + F(n-2) & if\ n \geq 2 \end{cases}$

其中 $+$ 表示字符串的连接。给出 n 和模式串 p，问 p 在 $f(n)$ 中出现了几次。

【数据范围】$n \leq 100$, $|p| \leq 10^6$, 答案 $< 2^{63}$ 。

2.4.2 算法分析

出现次数可递推 $f(n) = f(n-1) + f(n-2) + extra(n)$ ，其中 $extra(n)$ 表示 $F(n-1)$ 和 $F(n-2)$ 连接后新增的 p 的出现次数。而涉及到新增部分的长度不会超过 $2\,|p|$ ，所以在预处理时保留 $F(n)$ 的前 10^6 位和后 10^6 位即可，然后在计算 $extra(n)$ 时把涉及到的串拼接出来后用 KMP 算法求 p 在其中出现的次数即可。

【时间复杂度】$O(n \times |p|)$，【空间复杂度】$O(n \times |p|)$。

2.5 [E]Infiltration

2.5.1 题意简述

n 个点的图，两两之间有且仅有一条有向边。选出这个图的一个大小最小的点集，使所有点都能从点集中某个点经过最多一条边到达。

【数据范围】$n \leq 75$。

2.5.2 算法分析

由于两两都有边，所以对于当前未覆盖的点集，肯定可以选出一个点使未覆盖点集大小减少一半。因此使用贪心算法构造出的解答案最多$\log_2 n$。在此基础上，已经确定答案不大，所以使用搜索，枚举所有可能的选择方式，并加入最优性剪枝，使用 *bitset* 优化是否覆盖所有点的判断后即可通过数据。

【时间复杂度】$O(\sum_{i=1}^{\log_2 n} \binom{n}{i} \times \frac{n}{32})$，【空间复杂度】$O(\frac{n^2}{32})$。

2.6 [F]Keys

2.6.1 题意简述

有 n 个钥匙（分为两类）和 m 个钥匙环，给出初始的钥匙与环、环与环的连接状态，保证环与环的连接不会构成环。求最少的钥匙滑进滑出环的次数把两类钥匙分类（不同类钥匙不能在同一个钥匙环的块中），在这前提下最小化环滑进滑出的次数。

【数据范围】$n, m \leq 26$。

2.6.2 算法分析

钥匙要尽量少地移动，而只有在一个钥匙环上有两种钥匙时才需要移动，所以直接枚举每个钥匙环上留下的钥匙。但可能会出现所有环上都保留了某一种钥匙的情况，这时要腾出一个环给另一种，则枚举那个环，把它改为保留另一种。所有情况取最优值。

然后求最少的环操作，问题变为森林中若干点必须为1，若干点必须为2，还有一些点无要求，断开一些边，使所有不同类的钥匙分离，最后要把同类钥匙都连起来。这相当于要最小化断开的边数，树形DP， $f[i][j]$ 表示第 i 个点为根的子树中，当前保留的是第 j 类钥匙，断开的最少边数。

【时间复杂度】$O(2^{\frac{n}{2}} \times n^2)$，【空间复杂度】$O(n)$。

2.7 [G]Minimum Cost Flow

2.7.1 题意简述

三维空间内 n 个点，每个点上有 c_i 个洞，可以把洞作为连接口和其他点连接，费用为两点间的欧几里德距离，否则如果水到达了这个点，就必须把洞全部堵上，费用为每个洞 0.5。现在要从 1 号点向 n 号点运水，可以控制水平面高度，求最小费用。

【数据范围】 $n \leq 400$, $m \leq 50000$ 。

2.7.2 算法分析

枚举水平面的高度，则高度 $\leq$ 水平面的点构成了多个连通块。新建水管可以看作选一条从 1 号点所在连通块到 n 号点所在连通块的路径，要求路径上除了起始块和结束块之外的每个连通块都要有两个至少两个洞，而起始块和结束块也要有至少一个洞，费用为(路径上所有连通块的洞的个数之和-新建水管数×2) $\times$ 0.5。而找这样的路径相当于单源最短路问题，用并查集合并连通块后使用spfa或dijkstra算法即可。

【时间复杂度】$O(n \times spfa(V = n, E = n^2))$，【空间复杂度】$O(n+m)$

2.8 [H]Room Service

2.8.1 题意简述

从凸 n 边形内一点出发，碰到所有边至少一次后回到该点，求最短距离。碰到端点视为两条边都碰到。

【数据范围】$n \leq 100$。

2.8.2 算法分析

首先考虑这样一个问题，从点 A 出发，碰到直线 L 上任意一点后，到达点 B，求最短路径。做法是把点 A 沿直线做镜面反射，A' 与 B 的距离就是最短距离，连线就是方案。如果有两条直线，那就反射两次。但当 L 为线段时，交点不能取到，这时肯定是某个端点最优。

回到本题，有个比较显然的结论，最优方案中路径是不会相交的，一定是按照某个顺序依次经过每条边。又因为路径是可逆的，所以我们可以假设是按顺时针顺序。

而由开始的结论，只有端点的状态有效，先预处理端点两两之间的距离 $d[i][j]$，即把出发点沿中途经过边依次反射，最后得到的点和到达点构成的直线是最优路径。但该路径只有和每条反射的边都有交时才合法，所以不断倒着反射回去，看每条边是否和路径都相交。然后使用 $floyd$ 得到两点间最短路。

然后先枚举起始边，计算起点直接回到起点的方案，并预处理起点出发到每个点的距离 $Ds[]$ 和每个点出发到终点的距离 $Dt[]$，然后枚举起点出发到达的点 j 和出发到达终点的点 k（j 和 k 可以相等），用 $Ds[j]+d[j][k]+Dt[k]$ 更新答案。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

2.9 [I]A Safe Bet

2.9.1 题意简述

底面为$R\times C$的盒子中有 m 面垂直放置，水平呈‘/’状 45°放置的镜子和 n 面呈‘\’状 45°放置的镜子，从 $(1,0)$ 向右发射一条光线，若光线遇到镜子会发生90° 反射，若最后光线从 (R,C) 向右射出，则可以打开盒子。判断当前情况是否直接打开箱子，若不能，求出放置一面镜子使其打开的方案数（同一格两种放置方式只算一种），并输出字典序最小的一个答案。

【数据范围】$n,m\le 10^5$, $R,C\le 10^6$。

2.9.2 算法分析

先模拟光线的运动，记录发射位置和方向，每次在当前方向上找下一面镜子（用 *set* 优化），然后修改位置和方向，继续模拟。如果光线从 (R, C) 向右射出，则打开。否则从 $(R, C+1)$ 向左射出光线，反向追踪。可以发现可行位置为两次追踪得到的轨迹的交点个数。那么在两次追踪过程中分别记录下横向和纵向轨迹，然后第一次的横向和第二次的纵向求交，第二次横向和第一次纵向求交，求交过程可以用扫描线，并用线段树或树状数组优化单点修改，区间询问。求出交点个数和字典序最小的交点即可。

【时间复杂度】$O((n+m)\log C)$，【空间复杂度】$O(n+m+C)$。

2.10 [J]Shortest Flight Path

2.10.1 题意简述

地球表面有 n 个机场，要求从机场 s 飞到机场 t，且始终满足：离最近机场的距离不超过 R。由于油箱限制，最大连续飞行距离为 c。

本题距离都是指球面距离（假定飞机沿着地球表面飞行）。假设地球是半径为 $6370km$ 的球，有 Q 组询问 (s, t, c)。

【数据范围】$n \leq 25, Q \leq 100$。

2.10.2 算法分析

两个保护圈的交点是关键点，先两两求交求出所有关键点。

把机场和关键点当作节点，然后判断所有节点之间的路径是否合法，即判断大圆圆弧是否被所有保护圈的并所覆盖（对每个保护圈求出在大圆上对应的圆弧，转为区间覆盖问题）。

然后 $floyd$ 求出机场两两之间的最短路。对于每个询问，根据 c 重新判断每对机场是否可以直达。然后再做一遍 $floyd$ 后可得到最短距离。

下面考虑保护圈求交。R 是圆面距离，可以转为弧度，再转为空间距离，同时利用经纬度可以求出所有机场坐标和新的圆心坐标。

由于所有圆半径相等，则交点肯定位于与两圆圆心连线垂直的大圆面上。两圆心连线中点为该平面上另一点，利用这些可得到该平面法向量。

然后转化为一个圆和大圆求交，而这也是区间覆盖时需要使用的。

将小圆投影到大圆面上，设小圆圆心到大圆面的距离为 D，则两个交点到投影后的圆心的距离变为 $\sqrt{R^2 - D^2}$，利用该等式可重新建立一个在大圆面上的圆，然后变成了同一平面上的两圆求交。

先求出该平面上的 x, y 单位向量，然后套用平面几何的方法，余弦定理求出两圆心和某一交点组成的三角形中角 O 的大小，然后可得到两交点坐标。

而在圆弧覆盖时，需要确定一个 0 弧度的位置，然后求出其他弧度，可把起点设为 0 弧度，利用法向量可以求出 x, y 单位向量，其他求交时得到的圆心需要转换为相对弧度，即表示成 $(Ax + By)$ 的形式（注意有三个方程，但有两个是等价的，需要判断出来），然后用 $atan2$ 求弧度。

在 *LiveArchive* 上需要 long double 才能过。

最后还有一个要注意的是，两点在两极，则三点共线，无法求法向量。如果是两圆求交，那么求交无意义（交点不是关键点或没有交点），直接退出。如果是在判断路径是否合法时，只有 R 可直接覆盖全球时能直接判为可行，否则，我代码中直接判成不可行了（可能数据中没有这种情况）。但似乎不能直接判为不可行，可行路径不一定可以被表示为经过若干中转点得到的路径，所以需要把所有关键点所在大圆都判一遍。

【时间复杂度】$O(n^6 + Qn^3)$，【空间复杂度】$O(n^2)$。

2.11 [K]Stacking Plates

2.11.1 题意简述

n 个长为 m_i 的非递减序列合并为一个非递减序列，求最少的一分为二或 合二为一次数。

【数据范围】$n, m_i \leq 50$。

2.11.2 算法分析

如果没有相同元素，则排序后相邻元素来自不同序列的数量即为所需的合并次数。现考虑相同元素，则用 DP 来最小化合并次数，用 $f[i][j]$ 表

示高度 $\leq i$ 的，最后一个元素来自序列 j 的最小合并次数，转移时对于当前高度，由于来自相同序列的元素肯定连续最优，所以分别枚举第一个和最后一个元素来自的序列并转移。使用滚动数组更加方便。

【时间复杂度】$O(n^2m)$，【空间复杂度】$O(n \times m)$。

2.12 [L]Takeover Wars

2.12.1 题意简述

A 和 B 各有一个数字集合（大小分别为 n, m），两人轮流操作，A先手，每次操作方可以合并自己集合中两个数（值相加），或用己方集合中的一个值删去对方集合中的一个值（要求己方的值大于对方的值）。若某方集合为空则判输。

【数据范围】$n, m \leq 10^5$。

2.12.2 算法分析

首先发现两人都要让最大值越大越好，所以合并肯定是优先合并最大值和次大值，而删对方的值一定是用最大值去删除对方尽可能大的值。

所以如果你这一步能得到最大值比对方下一步能得到的最大值大，那么就选择合并，因为这一步如果删去对方的值，还不如下一步删。

否则如果能删去对方最大值则删去，因为如果对方删我的，对方的损失肯定比我的大，所以对方一定会合并，如果对方合并后最大值没超过我，显然比我合并后被对方超过好；如果对方合并后最大值会超过我，则说明即使我这一步合并也会被超过，所以还不如删去对方最大值。

如果我不能删去对方最大值，我删其他值损失肯定比对方大，所以只能选择合并。

把读入数排序后按这个策略选择下去即可。

【时间复杂度】$O(n \log n + m \log m)$，【空间复杂度】$O(n + m)$。

Chapter 3

ACM/ICPC World Finals 2011 (11/11)

3.1 [A]To Add or to Multiply

3.1.1 题意简述

写一个在长度尽量短的前提下字典序尽量小的包含 A 和 M 的程序，使输入 $x \in [p,q]$ 的输出 $\in [r,s]$，其中 A 表示 $x+=a$， M 表示 $x\times=m$。

【数据范围】$a,m,p,q,r,s \leq 10^9$, $p \leq q, r \leq s$。

3.1.2 算法分析

可以发现 M 的次数不会超过 $\log_m s$ (若 $m=1$ 则不需要 M)。那么枚举 M 的次数为 k，则 M 之间有 $k+1$ 个间隔，在不同间隔中放入 A 的效果不同，分别为 $+a\times m^i$ ，所以相当于求满足 $\sum_{i=0}^{k} a\times c_i m^i \in [r-p\times m^k, s-q\times m^k]$ 的 c_i 中使得程序长度最短的一组，最后比较 k 不同时的序列长度和字典序即可。

该条件可以转化为 $\sum_{i=0}^{k} a\times c_i m^i \in [\frac{(r-p\times m^k)}{a}, \frac{s-q\times m^k}{a}]$ ，把这个区间定义为 $[u,v]$ 。

然后只需贪心即可，从高位到低位，在不超过 v 的前提下能加就加，直到满足 $\geq u$。

在比较字典序时注意不是从最高位开始比，而是从第 0 个间隔开始比。

【时间复杂度】$O(\log_m^2 s)$，【空间复杂度】$O(\log_m s)$。

3.2 [B]Affine Mess

3.2.1 题意简述

给出平面上 3 个点，已知其经过旋转、缩放、平移或旋转、平移、缩放后变为了之后的 3 个点。其中旋转操作只能选取以原点为中心，边长为 20 的正方形边上的点作为旋转后 x 轴上点，且 3 个点经过旋转后会吸附到最近的网格(x, y 分别 $round$, $-n + 0.5$ 会变为 $-n$)。缩放操作只能选择非 0 整数作为缩放比例， x 和 y 的缩放比例可以不同。平移操作无限制。求解的情况（无解、唯一解、多解)。若整个平面经过两种变化后情况仍相同，这两种变化视为同一种。

【数据范围】$x_i, y_i \leq 100$。

3.2.2 算法分析

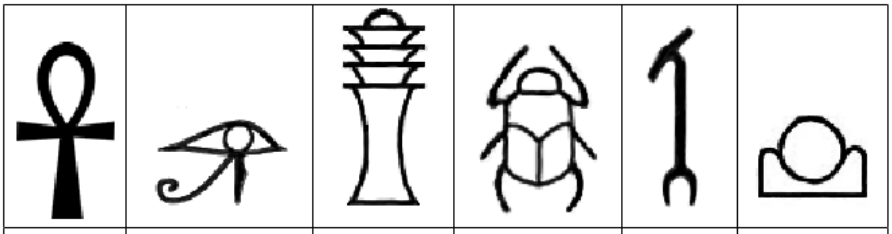
首先枚举旋转的方向，然后向量转角度得到旋转后的坐标，并吸附到网格。为防止重复统计，规定先缩放，再平移（由于缩放为非 0 整数，先平移再缩放可以对应到一种先缩放再平移的方案)，且规定缩放时 x 的比例不能为负数（ $-x, -y$ 和 x, y 会重复， $-x, y$ 和 $-y, x$ 会重复。由于不为负数较容易处理，在处理 y 时也可以强制不能为正，然后强制 y 取负数，再做一遍即可)。在规定比例为正后对应关系可以通过对点排序后确定，最后需讨论重复坐标的情况。

【时间复杂度】$O(80n)$，【空间复杂度】$O(n)$。

3.3 [C]Ancient Messages

3.3.1 题意简述

$n \times m$ 16 进制压缩的黑白图片，要求识别出以下 6 种图案。



Ankh	Wedjat	Djed	Scarab	Was	Akeht

【数据范围】$n, m \leq 200$。

3.3.2 算法分析

观察图片，发现每个图案内部连通块个数分别为 $0 \sim 5$，则先遍历最外侧空白，然后遍历出每个图案并编号，最后遍历内部空白，并确定包在外面的连通块编号，累加内部空白数量即可。

【时间复杂度】$O(nm)$，【空间复杂度】$O(nm)$。

3.4 [D]Chips Challenge

3.4.1 题意简述

$n \times n$ 的网格中，若干格子中不能放 C，若干格子中可以放 C，剩下的已经放了 C，要求在网格中再放下尽可能多的 C，使第 i 行和第 i 列的 C 的个数相同。且每行每列 C 的个数均不超过 C 的总数的 $\frac{A}{B}$。

【数据范围】$n \leq 40$, $1 \leq B \leq 1000$, $0 \leq A \leq B$。

3.4.2 算法分析

设第 i 行第 j 列放 C 则 $a_{i,j} = 1$，否则 $a_{i,j} = 0$，每行每列的限制为 L。对于所有 i，满足 $\sum_j a_{i,j} = \sum_j a_{j,i}$，转化为$\sum_j a_{i,j} - F_i = 0, F_i - \sum j a_{j,i} = 0$, $F_i \leq L$，可以发现所有变量在这 $2n$ 个等式中都出现了两次，且系数为

$+1$ 和 -1，而这些等式刚好对应网络流的流量平衡等式。则从 $+1$ 系数所在等式对应点向 -1 系数所在等式对应点连边，上下限由变量的限制确定。这样构造出一个循环流，且在 F_i 对应边上加上费用 -1，则要求最小费用可行流。强制负圈满流，强制下限，并添加源和汇，得到新图，在该图上求最小费用最大流即可。

最后，由于 L 还未确定，所以需要从 n 开始向下枚举 L，找到最多放置 C 的个数 $\times\frac{A}{B} \geq L$ 时 $break$。

【时间复杂度】$O(n \times CostFlow(n, n^2))$，【空间复杂度】$O(n^2)$。

3.5 [E]Coffee Central

3.5.1 题意简述

$n \times m$的地图上有 k 个特殊点。Q 个询问，定义 $S_{i,j}$ 为与 (i, j) 的曼哈顿距离在 r 以内的特殊点数量，询问 $\max\{S_{i,j}\}$ 以及对应的 (i, j)（$S_{i,j}$ 一样时选 j 最小的，也一样选 i 最小的）。

【数据范围】$k \leq 5 \times 10^5$，$Q \leq 20$，$n, m \leq 1000$。

3.5.2 算法分析

转换坐标 $(x, y) \to (x + y, x - y)$ 后变为求正方形交集。标记修改点后（即 $(x'-r_i, y'-r_i)+=1, (x'+r_i+1, y'+r_i+1)+=1, (x'-r_i, y'+r_i+1)-=1, (x'+r_i+1, y'-r_i)-=1$），前缀和处理即可。

【时间复杂度】$O(Q \times (nm + k))$，【空间复杂度】$O(nm + k)$。

3.6 [F]Machine Works

3.6.1 题意简述

有 n 种机器，第 i 种只能恰好在第 D_i 天购买，买入价格 P_i，卖出价格 R_i，使用该机器可每天盈利 G_i 元。初始 C 元，无机器，在 D 天内经营机器厂，使其获利最多。同一时间最多只能有一台机器，买入后和卖出前都需要停一天，可在同一天卖出、买入。

【数据范围】$n \leq 10^5$, $C, D, D_i, P_i, R_i, G_i \leq 10^9$。

3.6.2 算法分析

首先经营期间换机器肯定是为了增加日收益，所以机器只会越换越好，则先按 G_i 对所有机器排序。用 $f[i]$ 表示买到第 i 个机器时最多剩余的钱，则得到 $O(n^2)$ 的 DP （如果从买入第 j 台机器开始经过 $D_i - D_j$ 天的经营可以买入第 i 台机器，则转移）。然后可以发现要求的其实是**第** $D_i - 1$ **天**时最多能得到的钱。而钱关于时间是很多个一次函数，且只有下凸壳上的函数有用。那么接下来就是维护下凸壳，有因为斜率 G_i 递增，所以用栈维护即可（判断交点的前后）。要查询 D_i 天最多的钱数，可以通过下凸壳上交点横坐标二分到取到最大值的函数。然后计算并维护即可。

【时间复杂度】$O(n \log n)$，【空间复杂度】$O(n)$。

3.7 [G]Magic Sticks

3.7.1 题意简述

按顺序把 n 条线段分组，每组围成一个多边形，最大化总的面积和。

【数据范围】$n \leq 100$, $a_i \leq 1000$。

3.7.2 算法分析

假设已知分组情况，计算每组的最大面积，根据周长固定时圆的面积最大可猜想（并证明）这些线段端点在一个圆上时围成最大面积，则二分半径 $check$ 。需要注意圆心在多边形外的情况。这样就可以 $O(n^3 \log) DP$ 了，但会 TLE ，所以要优化，如果把一个完整的多边形分成两组，去掉最长的边最优。而如果已经能围成圆心在多边形内的多边形则不必再进一步分裂，这样就把要求的区间数降为 $O(n)$ 了，复杂度就降到了 $O(n^2 \log)$ 。注意多边形退化时二分不出半径，需要特判。非常卡精度，对不同情况需要开不同精度。

【时间复杂度】$O(n^2 \log)$，【空间复杂度】$O(n)$。

3.8 [H]Mining Your Own Business

3.8.1 题意简述

n 个点，m 条边的无向图中，最少确定几个特殊点，使任意一点在禁止经过后，所有点都能到达至少一个特殊点。

【数据范围】$n \leq 50000$。

3.8.2 算法分析

显然双连通分量可以缩成一个点，一个缩成的点中最多需要确定一个特殊点，然后就构成一颗无根树，只有树的叶节点才是必须确定为特殊点的。但注意只有一个双连通分量时，若特殊点禁止经过，则方案非法。所以至少建立两个特殊点。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

3.9 [I]Mummy Madness

3.9.1 题意简述

初始你在原点，有 n 个木乃伊分别在 (x_i, y_i)。每一时刻你可以向相邻 8 格走一步，然后所有木乃伊也会向相邻 8 格走一步，并使他与你的欧几里得距离尽量小（假设你与木乃伊都站在格子的中心位置）。问最多经过多少时间你会被木乃伊抓住，或永远也不会被抓住。

【数据范围】$n \leq 10^5$, $|x_i|, |y_i| \leq 10^6$。

3.9.2 算法分析

如果经过 10^6 的时间你还没有被抓住，那么你就永远不会被抓住。所以可以二分答案，问题转化为你在时间 T 内可以到的区域是否都能被木乃伊到达。这可以用扫描线和线段树来完成，但此题的正方形边长都一样，所以离散后的一段只需要两个中心最靠近 x 轴的正方形来覆盖，所以只需记录 x 轴上方和下方距离 x 轴最近的 y 坐标即可，即支持加减和询问最小值操作，这个可以用树状数组实现。

tsinsen 上被卡常数，需要在 *check* 时把肯定不会有影响的正方形去掉才能通过数据。

【时间复杂度】$O(n\log^2 n)$，【空间复杂度】$O(n)$。

3.10 [J]Pyramids

3.10.1 题意简述

相邻层边长差为 1 的是高金字塔，差为 2 的是矮金字塔，每层石块数量为边长^2 。要求把 n 个石块建造成个数尽量少的金字塔，且不能有两个金字塔相同，且在此基础上，方块数最多的要尽量多，次多的也要尽量多，依次类推。如果没有合法方案输出 *impossible*。

【数据范围】$n \leq 10^6$。

3.10.2 算法分析

先预处理出所有需要方块数$\leq n$ 的金字塔，金字塔数量是 $O(\sqrt[3]{n})$ 级别的。然后做一遍 01 背包得出最少需要的金字塔数，发现有解的答案最大为 6 。

然后用 $f[j][k]$ 表示总方块数为 j ，组成的 k 个金字塔中方块数最多的最少是多少。*LiveArchive* 上时限较紧，注意背包时最好把金字塔按所用石块数排序，这样可以简化计算，减小常数。并要注意减少冗余计算。

在询问时先得出金字塔数，然后每次都找到最大的金字塔，使其小于前一个，大于下一个，且建成它后剩余石块仍有合法方案（利用预处理的 f 直接判断）。

此外，此题直接暴力搜索可过。

【时间复杂度】$O(n^{\frac{4}{3}} \times 6)$，【空间复杂度】$O(n \times 6)$。

3.11 [K]Trash Removal

3.11.1 题意简述

二维平面中，把一个 n 个点的多边形放入平行管道中，求管道宽度最

小值。

【数据范围】$n \leq 100$。

3.11.2 算法分析

可以发现最后一定是某两点连线与管道平行。则枚举一点，求出这个点的最外侧点（若该点在内部则 *continue* ），两点连线卡住边界。然后用叉积求高度找出离这条线最远的点即可。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

Chapter 4

ACM/ICPC World Finals 2010 (11/11)

4.1 [A]APL Lives!

4.1.1 题意简述

表达式求值，其中变量的值可为 $1 \sim 3$ 维的向量，要支持 $+$ $-$ $\times$、赋值运算，同时有构造 $1 \sim n$ 的一维向量、删去向量前几个元素，以及修改变量的维数、每维大小的运算。运算无优先级，从右向左的顺序，可用括号改变计算顺序。

【数据范围】变量内元素个数 $(S) \leq 1000$ ， $-1000 \leq$ 元素的值 ≤ 1000 ，表达式个数≤ 10000。

4.1.2 算法分析

按题意模拟每种计算，用一个结构体来维护变量，同时记录维数和每维大小，可用 deque 来存储每个元素的值，并支持上述运算。注意处理表达式求值时要把相邻的常数合并为一维向量。

【时间复杂度】O(表达式个数 $\times S$)，【空间复杂度】O(变量数 $\times S$)。

4.2 [B]Barcodes

4.2.1 题意简述

$Code-11$ 的编码会独立地编码每一个字符（包括检验字符和开头结尾标志）为六位 2 进制数（第六位一定为 0）。对应关系表格和检验字符的计算方法见原题。

二进制信息用条形的宽度表示，宽的表示 1，窄的表示 0，并且宽的区域的宽度应该是窄的区域的宽度的两倍。

给出编码后的条形码 n 块区域的宽度，但读取宽度和实际宽度有 5% 的误差。

解码这个条形码，或指出条形码的错误。保证编码前条形码长度非 0。

【数据范围】$n \leq 150$, 所有的宽度≤ 250。

4.2.2 算法分析

首先识别 01 ，把所有宽度排序后，枚举长宽的分界线，则把窄的部分都 $\times 2$ 后也当作宽的，然后统计 min 和 max ，如果 $\min \times 105 \geq \max \times 95$ 则合法。

然后根据第二位判断条形码的方向，最后六位一组二进制压缩后按题意解压即可。需要注意的是每组第六位必须是 0 ，不要忘记判。

【时间复杂度】$O(n^2)$,【空间复杂度】$O(n)$。

4.3 [C]Tracking Bio-bots

4.3.1 题意简述

$m \times n$的地图中有 w 堵横向的墙，问图中有多少格子不能只向右上走到右上角。

【数据范围】$m, n \leq 10^6$, $w \leq 10^3$。

4.3.2 算法分析

离散后从右上角向左下方递推即可，统计不能达到的格子数。

【时间复杂度】$O(w^2)$，【空间复杂度】$O(w^2)$。

4.4 [D]Castles

4.4.1 题意简述

n 个城堡构成的树，进攻每个城堡至少需要 a_i 个人，会减少 b_i 个人。现在以任意一个城堡作为第一个进攻的城堡，随后依次攻占所有城堡，军队在同一方向上经过同一条边最多一次，要求最小化所需要的人数。

【数据范围】$n \leq 100$。

4.4.2 算法分析

如果可以按任意顺序攻占城堡，则按 $b_i - a_i$ 的顺序排序后依次攻占即可。因为总所需人数为 $\max\{\sum_{i=1}^{j-1} b_i + a_j\}$，考虑相邻两人 i, j，要满足 $\max(a_i, b_i + a_j) \leq \max(a_j, b_j + a_i)$，所以 $b_i - a_i \leq b_j - a_j$。

而在树上时，由于一条边同一方向只能经过一次，所以相当于先算出每个子树的 f_i, g_i，并决定每棵子树的顺序，把所有子树按 $g_j - f_j$ 排序即可。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n)$。

4.5 [E]Channel

4.5.1 题意简述

在 $n \times m$ 的带障碍地图上，求一条最长的从左上角出发到右下角的路径，且路径不能自交（除路径上相邻两格外无公共点）。输出方案，保证唯一解。

【数据范围】$n \leq 20, m \leq 9$。

4.5.2 算法分析

很明显是连通性状态压缩DP。但由于对角线的也有影响，轮廓线需要加一个状态，即当前格的左上角。而且为了判断自交，如果某方向没有插

头，需要知道那格是否在路径上，所以需要 4 进制表示状态：空，在路径上，‘(’插头，‘)’插头。而由于起点和终点在边界上，可以认为边界外有一个‘(’插头，把该路径当成‘)’插头，不需要设成独立插头。和原来的转移相比，要注意若左和上方向均为 1 ，则不能转移；一旦（左或上方向有 1 状态）或（左上为 1 ，且左和上均为 0 ），则该格必须为 0 ；‘(’和‘)’不可合并。下一次转移的左上状态可直接由当前上方状态得到。由于限制较多，每层的状态数不会太多，开个数组进行重编号后就能存下每层状态了，然后记录下转移过来的状态以输出方案。

【时间复杂度】$O(nm4^{m+2})$，【空间复杂度】$O(nm4^{m+2})$。

4.6 [F]Contour Mapping

4.6.1 题意简述

n行正三角形点阵，奇数行点数为m，偶数行点数为m+1，每点高度为 $h_{i,j}$。正三角形边长为 d ，正三角形区域被视为平面，由其三个顶点的坐标和高度确定。现要每隔 h 的高度画若干等高线。求等高线的总长度。注意一块区域水平时，只有区域边界有等高线。

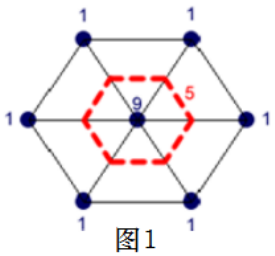


图1

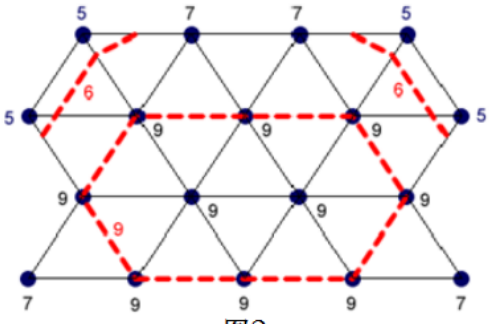


图2

【数据范围】$n, m \leq 100$, $d \leq 10$, $h \leq 1000$, $0 \leq h_{i,j} \leq 10^6$。

4.6.2 算法分析

如果等高线不是沿着正三角形的边，那么不会出现一整块水平的情况，每个三角形可以单独计算。

对于一个正三角形，可以按中间高度的线割成两个三角形计算，并用余弦定理算出割开的线的长度。

可以发现一个三角形中的等高线是均匀分布的，长度构成了一个等差数列，得到首项末项算出项数求出和，利用相似换算成等高线长度即可。

而沿着正三角形的边的等高线，只有在这条边两端高度相等且是 h 的倍数时，且包含这条边的两个正三角形的另一顶点中，至少一个和这条边两端点的高度不同时，这条边在等高线上。

【时间复杂度】$O(nm)$，【空间复杂度】$O(nm)$。

4.7 [G]The Islands

4.7.1 题意简述

二维平面上 n 个 x 坐标互不相同的点，从最左侧点一直向右走到最右侧点，再一直向左走回到最左侧点，要求除了最左侧点外所有点都被经过恰好一次，且有两个特殊点必须在不同的阶段经过，求最短的总路程和字典序最小的解。

【数据范围】$n \leq 100$, $0 \leq x_i, y_i \leq 2000$。

4.7.2 算法分析

可以看作两次都是从最左走到最右，则用 $f[i][j]$ 表示第一次走到 i，第二次走到 j 的最小路程。讨论 $\max(i, j) + 1$ 在哪一次经过即可，并记录转移来源以输出方案。而对于特殊点，由于路径可翻转，可以强制第一次经过第一个特殊点，而第二次经过第二次特殊点，如果方案字典序不是最小则翻转方案即可。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n^2)$。

4.8 [H]Rain

4.8.1 题意简述

给出 n 个高度为 h_i 个点 m 条边的连通平面图，且所有区域均为三角形，边界外的点高度都比与其相邻的边界低。求暴雨淹没整个区域后形成的湖泊个数。两个只有一些深度为 0 的公共点的湖泊会被当作不同的湖泊。

【数据范围】$n \leq 52^2$。

4.8.2 算法分析

一个点的水面高度取决于它到边界需要经过的海拔最高的点。则从边界上的点出发，$dijkstra$ 求出每个点的水面高度，然后对于有水的部分 dfs 或 bfs 求连通块即可。

需要注意的是求边界的问题，可以先找出 x 坐标最小的点，该点肯定在边界上。从该点出发的极角最小的边肯定是边界。从该边出发并绕回该边（边界会经过一个点多次，若从点出发会使边界不全），每次往逆时针方向第一条边走，即反向边的下一条边。可以使用循环链表（或加链尾特判，比如链尾指向某一负数）实现，将所有边排序后依次加入链表以保证有序，链表节点从 2 开始标号后 xor 1 即可得到反向边。本题范围较小，可以暴力找反向边（会被卡到 $O(m^2)$，利用单调性后为 $O(m)$）。

【时间复杂度】$O(m \log m)$，【空间复杂度】$O(m)$。

4.9 [I]Robots on Ice

4.9.1 题意简述

$n \times m$ 的图中，从 $(0,0)$ 开始遍历图，到 $(0,1)$ 结束，要求读入的三点分别为路径的 $\frac{1}{4}$、$\frac{1}{2}$、$\frac{3}{4}$ 处（下取整）。求方案数。

【数据范围】$n, m \leq 8$。

4.9.2 算法分析

由于有路径上三个点的限制，所以只能搜索。首先可以剪枝的是如果下一阶段的点到当前点的曼哈顿距离+当前步数已超过目标步数，则剪枝。其次读入点和终点只能在确定的步数时经过，其他时间都不可经过。最后由于要遍历整个图，所以路径不能把图断为多个连通块，如果每次都floodfill暴力判断的话反而会得不偿失，需要简化判断。可以发现如果当前点四周只有两个空格，且方向相反，那么两者一定已经被隔开了。还有如果两空格方向相邻，那么两空格的公共相邻格也必须为空，否则就被隔开了。加上这些剪枝后就可通过数据。

还有一个剪枝，就是当前点的相邻格中，如果该格不是终点，且其周围空格数为 1 ，那么下一步必须向那边走。如果有多于一个这样的点则无解。

【时间复杂度】$O(\ (nm)!\ )$，【空间复杂度】$O(nm)$。

4.10 [J]Sharing Chocolate

4.10.1 题意简述

问能否把 $x \times y$ 的矩形沿整数坐标切 $n-1$ 刀分为 n 块，且每块面积为 a_i。

【数据范围】$x, y \leq 100$, $n \leq 15$。

4.10.2 算法分析

$f[i][j]$ 表示二进制状态为 i 的块构成其中一边为 j 的矩形是否可行。转移即枚举一个子集进行划分，通过讨论切那条边可以得出划分后的矩形。

【时间复杂度】$O(3^n)$，【空间复杂度】$O(2^n \times x)$。

4.11 [K]Paperweight

4.11.1 题意简述

由两个拥有一个公共面的四面体拼接而成的物体内有一个特殊点。问在所有能稳定放置该物体的方案中，特殊点离底面的最近和最远距离是多少。其中稳定放置是指重心偏移 0.2 单位后物体仍不会倒。

【数据范围】点的每维坐标的绝对值都不超过 1000。

4.11.2 算法分析

物体会不会倒取决于重心的垂直投影是否落在支撑面内，所以首先要计算重心。

而四面体重心的坐标为四个顶点坐标的平均值，而这个物体由两个四面体组成，则重心为两个四面体重心按各自体积的加权平均。

然后就是枚举支撑面，求法向量，判断是否另两点不是异侧。然后看重心的投影是否在支撑面的三角形内（判断重心在三个垂直面的哪侧），并计算到三条边的距离。如果合法则用特殊点到当前支撑面的距离更新答案。

但是这样做是有问题的，比如一个金字塔形状的物体，底面四点共面，对于枚举的四个三角形，重心都落在边上，都不稳定。而其实这种情况是稳定的。所以在遇到四点共面时还要判断重心是否落在四边形内，可以默认四边形为凸四边形，因为凹四边形不会出现这种情况。

【时间复杂度】$O(5^3)$，【空间复杂度】$O(1)$。

Chapter 5

ACM/ICPC World Finals 2009 (11/11)

5.1 [A]A Careful Approach

5.1.1 题意简述

有 n 架飞机降落，第 i 架飞机必须在 $[a_i, b_i]$ 区间段内降落，求最长的最小时间间隔。

【数据范围】$n \leq 8$。

5.1.2 算法分析

二分答案，显然当前状态只和最迟降落的飞机的时间有关，而且这个值越早越好，则用 $f[i]$ 表示已经降落的飞机的二进制状态为 i，最迟降落的飞机最早降落的时间，转移为枚举下一个降落的飞机 j，根据其降落时间窗口，计算更新 $f[i|1 << j]$ 的值。

【时间复杂度】$O(n2^n \log)$，【空间复杂度】$O(2^n)$。

5.2 [B]My Bad

5.2.1 题意简述

给出逻辑电路的描述（ N 个输入， G 个门， U 个输出），但其中最多有一个门损坏了，变成取反原答案或者只输出0或者1。现在给出这组电路的 B 组输入输出，判断电路的情况。

【数据范围】$n \leq 8$, $G, U \leq 19$。

5.2.2 算法分析

由于 N 最多为 8 ，所以最多 2^8 个不同的输入。然后暴力枚举错误状态，并枚举每组输入，拓扑 $O(G)$ 或暴力 $O(G^2)$ 计算并和输出比对。

【时间复杂度】$O(2^N \times G^2)$，【空间复杂度】$O(G + 2^N)$。

5.3 [C]The return of Carl

5.3.1 题意简述

棱长全为 10 的正八面体上，给出起点和终点相对于中心的水平角度和垂直角度，求两点在表面上的最短距离。

【数据范围】水平角 $\in [0, 360)$ 垂直角 $\in [0, 180]$。

5.3.2 算法分析

利用两个角度直接求点在面上的相对位置，然后枚举路径穿越的上下两部分的交界边，分别摊开两部分，求出每个三角形的位置，然后利用相对位置直接把点放到平面上。其中两点间最短的距离就是答案。虽然某些方案中两点连线上没有展开的面，但在那种情况下肯定不是最优的，所以不需要考虑。

【时间复杂度】$O(4 \times 2^2)$，【空间复杂度】$O(1)$。

5.4 [D]Conduit Packing

5.4.1 题意简述

给出 4 个小圆的直径，求最小直径的大圆使 4 个小圆能不重叠地放入其中。

【数据范围】直径 ≤ 20000。

5.4.2 算法分析

二分答案，搜索圆的放置顺序判断是否可行，在依次放置过程中当前圆（第一个除外）肯定与之前至少两个圆（包括大圆）相切。

但其实所有小圆均和大圆相切的情况不会变劣，所以在搜索放置顺序后直接按顺序贴大圆边界放过去，最后判断总占用弧度是否超过 2π 即可。

【时间复杂度】$O(4!)$，【空间复杂度】$O(1)$。

5.5 [E]Fare and Balanced

5.5.1 题意简述

给 n 个点 m 条边的有向带权图的某些边增加边权，使 1 号点到 n 号点的所有路径权值和相同，且没有一条路径上增加了多次边权。

【数据范围】$n, m \leq 50000$。

5.5.2 算法分析

定义 $f(i,j)$ 为 i 到 j 的最大边权和，g(i,j)为最小边权和。

若图中存在一点，起点到该点和该点到终点的路径均不唯一，则说明两侧都需要增加边权，肯定有路径增加了多次边权，不合法。即当且仅当存在点 i 满足 $f(1,i) \neq g(1,i)$ 且 $f(i,n) \neq g(i,n)$ 时无解。

显然最后路径长度都改为 $f(1,n)$ 最优。对于每条边 (i,j)，若 $f(1,i) = g(1,i)$ 且 $f(1,j) = g(1,j)$ 或 $f(i,n) = g(i,n)$ 且 $f(j,n) = g(j,n)$，则说明此边边权不需要修改。只有在 $f(1,i) = g(1,i)$ 且 $f(1,j) \neq g(1,j)$ 时，而 $f(j,n) = g(j,n)$，此边必须修改，且需要使路径长度与最大的相等，即边

权要达到 $f(1,n)-f(1,i)-f(i,n)$。这样所有路径至多经过一条修改边，且所有路径的长度和相等。

上述算法中只用到了 $f(1,i),g(1,i),f(i,n),g(i,n)$，均可在拓扑排序后 DP 计算。

【时间复杂度】$O(n+m)$，【空间复杂度】$O(n+m)$。

5.6 [F]Deer-Proof Fence

5.6.1 题意简述

要求建造若干篱笆，使 n 棵树周围距离 m 以内的区域都在篱笆内。

【数据范围】$n \leq 9, m \leq 200$。

5.6.2 算法分析

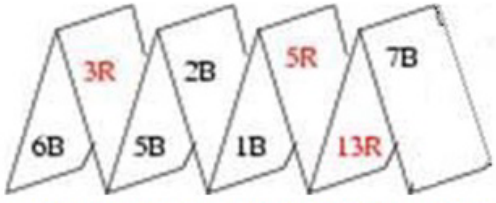
枚举包在同一个篱笆内的集合，计算费用，最小费用为半径均为m的圆的曲凸包的长度，即原来的点凸包长度+$2\pi \times m$（圆绕了一圈）。然后 DP 一遍，$f[i]$ 表示已包围子集为 i 的最小费用，转移为枚举下一个子集。

【时间复杂度】$O(3^n)$，【空间复杂度】$O(2^n)$。

5.7 [G]House of Cards

5.7.1 题意简述

两人用共 $2m$ 张互不相同的最大分数为 m 的红黑两色牌玩游戏。初始选出 8 张牌按如图方式组成 4 个“山峰”，第一张牌的颜色决定了先后手。



由牌堆的最上面 8 张牌组成的山峰和山谷

玩家每搭成一个三角形，就要给三角形中颜色较多的一种对应的玩家加上 3 张牌的分数和。当前玩家拿到序列中的下一张牌后，若该玩家还未持有牌，则他可以选择持有这张牌，或把这张牌放到某个“ 山谷”上。若他已持有牌，他可以选择任意一张放到“山谷”上，然后持有剩下那张，或者两张一起搭在“平地”上。等到纸牌全部拿完后游戏结束，此时持有在各玩家手上的纸牌需要给颜色对应方加上纸牌上的分数。现在两人都采用最优策略，问在给定的纸牌序列下，两人最多能赢或最少能输的分数。

【数据范围】$5 \leq m \leq 13$。

5.7.2 算法分析

搜索，可以用两人持有的牌和房子上实际有用的 6 张牌和形状（用 1 表示下降， 0 表示上升，其中“平地”也占用两个位置，其中第一个位置用 0 号牌来表示这是“平地”）来表示当前状态，而只有连续两张为 10 （“山谷”或“平地”）时才可以在上面放牌，分类讨论继续搜索（角色互换）。需要加 $\alpha - \beta$ 剪枝，即在当前搜到的最大值，已经不小于上一轮中对手能够得到的最小值，那么对手显然不会选择该分支，当前轮中的剩余的搜索都是无意义的，直接可以退出。

【时间复杂度】$O(4^{2m-8} \times 6)$，【空间复杂度】$O(2m)$。

5.8 [H]The Ministers' Major Mess

5.8.1 题意简述

m 个大臣给 n 个议案投票，每个大臣投了 k_i 票。要求所有大臣都要有大于一半的投票成立，求是否有可行方案，若可行，求出每个决策是必须被通过，还是否决，还是两者都可能。

【数据范围】$n \leq 100, m \leq 500, k \leq 4$。

5.8.2 算法分析

由于是大于一半通过，所以最多一个不通过，当 $k < 3$ 时就需要全部通过，则直接从该 *bill* 的非 *vote* 方向 *vote* 方连边。而 $k > 2$ 时，若某项不通

过，则其余必须通过，从该投票的某一 *bill* 的非 *vote* 方向其他 *bill* 的 *vote* 方连边。如果此图出现某议案的通过和否决在同一强联通分量中，则无解。否则枚举每个议案，由 2-*sat* 的对称性，分别看 通过能否到达否决 或 否决能否到达通过。由于点数很小，可直接使用 $floyd$。可以使用 *bitset* 优化 $floyd$。也可以每次从起点 bfs，看是否能到达终点。

【时间复杂度】$O(n^2)$　$O(\frac{n^3}{32})$，【空间复杂度】$O(n^2)$。

5.9 [I]Struts and Springs

5.9.1 题意简述

平面上有 $n+1$ 个窗户，当外层窗户的大小变化时，里面的窗户也会变化，而支杆和弹簧是决定里面的窗户怎样变化大小或改变位置的装置。每扇窗户覆盖了平面上的一个矩形区域，窗户可以包含别的窗户以产生层次。当最外面的窗户改变形状时，每一个被它直接包含的窗户会改变位置或大小（基于支杆和弹簧的布置）；这些变化可能又会导致更里面的窗户改变位置、大小。支杆连接部分长度固定，而弹簧连接部分可以按比例拉伸或压缩。给出最外面的窗户的大小以及 n 个内层窗户位置大小以及其内部或和直接包含它的窗户的连接的方式（支杆还是弹簧，顺序依次为连接垂直边、水平边，上、下、左、右方向的边与直接包含它的窗户的相对的边），对于 m 个改变最外面窗户大小的操作输出这个操作后所有内部窗户的位置和大小。保证缩放前后窗户都不会重叠，缩放时若遇到均为支杆不能缩放时，应用弹簧取代最右侧或最上侧的支杆。

【数据范围】$n, m \leq 100$, $Tsinsen$上是 ≤ 500。

5.9.2 算法分析

首先对于每个窗户，枚举其他所有窗户，求出父窗户，即直接包含它的窗户。缩放时按从外到内缩放，一个节点缩放完后处理它的所有子窗户。弹簧长度只需按 (总长 – 支杆总长) 按比例缩放。注意读入中 6 个支杆或弹簧的顺序。

【时间复杂度】$O(nm)$，【空间复杂度】$O(n)$。

5.10 [J]Subway Timing

5.10.1 题意简述

给出 n 个点的树，要求把每条边的边权改为 60 的倍数，使所有两点间变换后的距离和实际距离的差值的绝对值中，最大的差值的绝对值最小。

【数据范围】$n \leq 100$。

5.10.2 算法分析

先二分答案，$f[i][j]$ 表示 i 为根子树内，到节点 i 的差值区间右端点为 j 的且满足二分的答案的情况下，最靠右的差值区间左端点是多少。

在计算以 i 为根的子树时，枚举其子树 j（事先加上 (j,i) 这条边的权值），枚举两者的差值区间右端点，得到最优的左端点，若两个端值相加的绝对值不超过二分值，则可以转移。

但这样复杂度为 $O(60^2n^3)$，但可以先以构造较优解减小二分范围。总存在一种方案使所有点到其所有祖先的差值在 $[-59, 59]$ 以内，那么这样最后的差值就不会超过 118，这样就可以从 118 的上届开始二分。

【时间复杂度】$O(118^2 n \log 118)$，【空间复杂度】$O(118n)$。

5.11 [K]Suffix-Replacement Grammars

5.11.1 题意简述

给出 n 个后缀替换规则，判断从初始串最少经过几次变换才能变为目标串。

【数据范围】$n \leq 100$, 串长度 ≤ 20。

5.11.2 算法分析

按串长度分层，每层为所有为该长度的后缀，该层两点间，若首字母相同，则距离可由上一层对应节点的距离得到。同层之间 $floyd$ 更新。

【时间复杂度】$O($串长 $\times\, n^3)$，【空间复杂度】$O($串长 $\times\, n^2)$。

Chapter 6

ACM/ICPC World Finals 2008 (11/11)

6.1 [A]Air Conditioning Machinery

6.1.1 题意简述

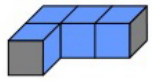


Figure 1

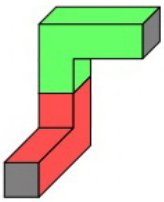


Figure 2

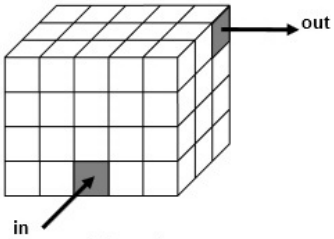


Figure 3

在 $n_x \times n_y \times n_z$ 空间内，有一个入水口和出水口（如 $Figure\,3$ ），你需要用尽量少的管道(如 $Figure\,1$)，用如 $Figure\,2$ 的方式拼接起来，并使水能从入水口流到出水口。若所需管道数超过 6 个输出 $Impossible$。

【数据范围】$1 \le x \le n_x \le 20, 1 \le y \le n_y \le 20, 1 \le z \le n_z \le 20$

6.1.2 算法分析

直接搜索，对于当前所在位置和水流方向，枚举下一个管道的放置方法（最多 8 个方向），记录空间内方格被占用情况，遇到非法情况退出。

如果把管道拆成4个独立方格依次处理，用一步步走来处理 8 个方向可以更方便些。

【时间复杂度】$O(8^6)$ ，【空间复杂度】$O(n_x \times n_y \times n_z)$。

6.2 [B]Always an integer

6.2.1 题意简述

给出多项式

$$\frac{\sum_{i=0}^{k} a_i x_i}{D}$$

问对于任意正整数 x ，多项式的值是否均为整数。

【数据范围】$k \leq 100$, a_i, D为整数。

6.2.2 算法分析

可以转化为 $\sum_{i=0}^{k} a_i x_i$ 是否总能被 D 整除。

如果 $f(x+1) - f(x) \equiv 0(mod\ D)$ 对于x为正整数恒成立 且 $f(1)$ 能被 D 整除，那么 $f(x)$ 总能被 D 成立。即我们只需验证 $f(1)$ 能否被 D 整除 和 $g(x) = f(x+1) - f(x)$ 是否总能被 D 整除，就能确定f(x)是否总能被 D 整除。而 $g(x)$ 是一个 $k-1$ 次多项式。依次类推到只有常数项，这时便可直接判断。所以对于一个 k 次多项式，需要判断 $k+1$ 个 $f(x)$ 能否被 D 整除。

所以只需验证 $x = 1 \rightarrow k+1$ 是否都能被 D 整除即可。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

6.3 [C]Conveyor Belt

6.3.1 题意简述

给出平面上的若干个圆形旋转轴，旋转方向为顺时针或逆时针，要用传送带连接某两个指定的轴，使得传送带尽可能短。直接连接的两个轴之间的传送带长度不超过给定的值 d（绕在轴上的长度无限制）。传送带不能自交，不能穿过轴的内部。

【数据范围】$n \leq 20$。

6.3.2 算法分析

先按照旋转方向，求任意两轴的对应公切线（用余弦公式求出切线平移后构成的三角形的角度，利用向量转角度可求得公切线），注意判断公切线长度超过 d 以及穿过某个轴内部的不合法情况。

然后开始搜索，记录下当前的路径，注意新增的边和之前的所有边都不能相交。可加最优性剪枝。

最恶心的是边界情况，当两条公切线只有一个交点时是合法的，中间的转轴作为临时中转是允许的。而公切线和转轴相切时是不合法的。不过这两种情况合起来就是三轮相切时，中间那个能否作为中转轮，而这样特判的结果是符合实际情况的。

【时间复杂度】$O(n!)$,【空间复杂度】$O(n^2)$。

6.4 [D]The Hare and the Hounds

6.4.1 题意简述

在 n 个点 m 条边的图上，猎狗要找到野兔走的一条非空路径。在非选择点，野兔和猎狗选择边都遵循主要道路规则“尽量直”，即转最小的角度（可能不需要转弯），如果两条边转的角度相同，选择右侧那条。在选择点，野兔可能违反主要道路规则（但不会去来的那条边），随机选一条路走，并在到达下一个选择点前放下确认标记。猎狗遇到选择点后，必须尝试每一条边，直到到达一个确认标记。

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到了死胡同（只有一条边的点）。

3. 猎狗在碰到确认点前到了其他的选择点（野兔总是会在到达下一个选择点的路上放确认点，且野兔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，即使中间到达了终点也会忽略。猎狗在选择点时会忽略所有已经走过的边（包括来的那条）后 按最后到达这个点的方向使用主要道路规则选择下一条边。猎狗不会记得它走向一条路的结果，即使他在寻找确认点的时候不断的经过这条路。

题目保证可以找到一条合法路径。输出野兔走的距离，猎狗走的距离（包括走错的），以及野兔依次经过的边。

【数据范围】$n \leq 100$, $m \leq 150$, 确认点个数 ≤ 100 , $maxdis \leq 2000$, 每个点不会有两条边从同一角度连出去。

6.4.2 算法分析

先预处理每条路上最近的确认点。然后开始搜索。

把寻路分为两种状态：普通状态和搜寻状态，普通状态直接模拟，直到遇到选择点进入搜寻状态，搜寻状态如果判为不正确则回溯，如果到达了确认点则回到普通状态。

需要注意的是路径要非空，起点和终点相同时要特判。

【时间复杂度】$O(n \times m \times maxdis)$ ，【空间复杂度】$O(n \times m)$。

6.5 [E]Huffman Codes

6.5.1 题意简述

给出一颗 n 个叶子节点的哈夫曼树（满足$\sum$叶节点权值×深度 最小），每个非叶节点的权值为左右儿子之和，要求左儿子权值≤右儿子权值，根的权值为 100。求叶节点权值分配的方案数。

【数据范围】$n \leq 20$。

6.5.2 算法分析

根据哈夫曼树的构造过程，可以发现同一层的节点的权值从左到右非递减；而对于不同层的两个节点，由于深度更深的点先发生合并，说明它的权值比另一个小。所以整棵树的权值满足从下到上，从左到右非递减。又由于 n 较小，限制较多，所以答案不会很大，直接搜索就能通过。

【时间复杂度】$O(\binom{100}{n})$，【空间复杂度】$O(n)$。

6.6 [F]Glenbow Museum

6.6.1 题意简述

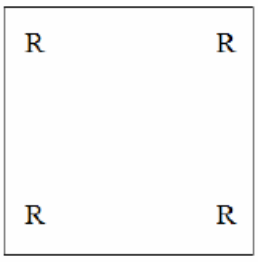


Figure 1: A rectangle

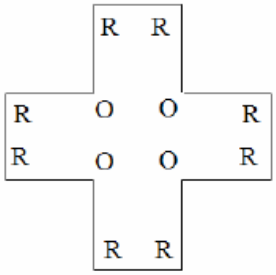


Figure 2: A cross-shaped polygon

用 $R(90°)$ 和 $O(270°)$ 表示一个直角多边形内角的度数，判断对于给定长度，有多少用 R 和 O 组成的序列能画出闭合的图形，并且在图形内部有一点能看到所有边界。

6.6.2 算法分析

因为多边形外角和为 $360°$，所以 R 的个数比 O 的个数多 4 个（将 $270°$ 视为 $-90°$），所以 L 为奇数或者小于 4 时无解。经过观察发现，一个合法的序列只要不包含 OO，一定可以画出符合条件的多边形。那么题目转化为求包含 $(L+4)/2$ 个 R 和 $(L-4)/2$ 个 O 的序列，满足首尾相连后不存

在 OO 的排列个数。考虑首位是否为 O，记 R 的个数为 K，那么答案就是 $\binom{K}{4}+\binom{K-1}{4}$。

当然也可以用 DP 求，$f[i][j][k]$ 表示前 i 个位置，R 比 O 多 j 个，且上一个位置放的是否是 O，由于 $R>5$ 已经肯定无解了，所以 j 的范围为 $[-1,5]$。转移分放 R 和 O 讨论即可。最后减去强制第一位和最后一位均为 O 的方案数即可。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

6.7 [G]Net Loss

6.7.1 题意简述

给出多项式

$$p(x)=\sum_{i=0}^{n}a_i x^i$$

定义

$$g(x)=\begin{cases} a_1x+a_0, & -1\le x\le c \\ b_1x+b_0, & c\le x\le 1 \end{cases}$$

$$d(p,g)=\int_{-1}^{1}(p(x)-g(x))^2\,dx$$

给出 $p(x)$，c，求最优的 a_1,a_0,b_1,b_0 使 $d(p,g)$ 最小。

【数据范围】$n\le 10$。

6.7.2 算法分析

当 $g(c)$ 确定时，两侧可分别考虑，把斜率当未知数，则 $d(p,g)$ 关于斜率为二次函数，得出系数后可直接得到最优解。

经过打表观察发现 $d(p,g)$ 关于 $g(c)$ 也是二次函数。

求未知二次函数 $f(x)=ax^2+bx+c$ 求极值当然可以三分，但还有一种很神的方法。

分别把 $x = -0.5, 0.5, 1.5$ 代入，则

$$f(-0.5) = 0.25a - 0.5b + c \tag{6.1}$$

$$f(0.5) = 0.25a + 0.5b + c \tag{6.2}$$

$$f(1.5) = 2.25a + 1.5b + c \tag{6.3}$$

(6.2)−(6.1)可得到 b，(6.3)−(6.2)可得到 $2a + b$，从而得到 $2a$，那么就可以得到 x 的最优取值 $-\frac{b}{2a}$ 了。

所以先求 $g(c) = -0.5, 0.5, 1.5$ 时 $d(p, g)$ 的最优值，最优值可通过 $a_1, b_1 = -0.5, 0.5, 1.5$ 求。从而得出最优的 $g(c)$，此时再求得 $g(c)$ 最优时的 a_1, b_1，得到方案。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

6.8 [H]Painter

6.8.1 题意简述

给出 n 个三角形，先判断是否有三角形相交，若没有三角形相交则求被三角形包含次数最多的区域被包含的次数 +1。

【数据范围】$n \leq 10^5$，$-10^5 \leq x_i, y_i \leq 10^5$。

6.8.2 算法分析

首先判断数据合法性，判断平面中是否存在相交线段是有经典做法的[1]。首先为了避免垂直线段的干扰，我们对点进行剪切变换 $(x, y) \rightarrow (x + y * eps, y)$。接下来就是沿 x 坐标扫描，遇到线段左端点则把线段插入 set 中，并与它上方和下方的线段分别求是否相交。而遇到线段右端点时把该线段从 set 中删除，并判断原来在它上方和下方的线段是否相交。简略证明的话就是两线段相交前在 set 中一定是相邻的（否则已经出现其他的相交），详细证明参见《算法导论》。

此题略有不同，即属于同一个三角形的三边允许相交，这个只需要在判断线段相交时特判掉来自同一三角形的情况即可。这样可以判断数

[1] 见《算法导论》 33.2 节 确定任意一对线段是否相交

据的合法性了。之后就是求包含的次数，仍然是扫描线的做法，对于当前的扫描线，我们发现一个三角形一定是两条边在 set 中，因此可以把它们看作左右括号，也就是三角形构成的树的 dfs 序。我们只要维护括号对应节点在树上的深度就可以取max得到最大的包含次数了，即最多需要的明暗度的数量。为了减少特判，我们初始把画布也加入set中，即线段 $(-10^7,-10^7),(10^7,-10^7)$ 和 $(-10^7,10^7),(10^7,10^7)$。在扫到三角形最左侧顶点时，我们确定该点连出去两条边的上下关系就可以确定左右括号关系了，并在插入时求该括号对应点的深度。若插入的是左括号，如果其左侧为左括号，说明左侧是它父亲，则该点深度 = 父亲深度 +1；如果左侧为右括号，说明左侧是它兄弟，则该点深度 = 父亲深度 +1。若插入的是右括号，类似地讨论即可。在扫到三角形中间的顶点时，删去左侧点到它的边，并得到这条边的左右括号属性，插入它到右侧点的边即可。扫到三角形最右侧顶点，则删去两边即可。在插入和删除的同时进行线段相交的判断。

【时间复杂度】$O(n\log n)$，【空间复杂度】$O(n)$。

6.9 [I]Password Suspects

6.9.1 题意简述

问包含 m 个给出串的长度为 n 的仅包含小写字母的串的数量。

【数据范围】$n\le 25, m\le 10$。

6.9.2 算法分析

先去掉被包含的串，然后构造 m 个串的 AC自动机，$f[i][j][k]$表示长度为 i，当前包含的串的二进制状态为 j，且在AC自动机上的 k 号节点的方案数。转移即枚举下一个字母，然后更新包含状态和到达节点即可。

对于输出方案，由于要求输出数量较少。直接从结束状态出发，枚举转移过来的状态即可。

【时间复杂度】$O(26nm\left|s_i\right|\times 2^m)$，【空间复杂度】$O(nm\left|s_i\right|\times 2^m)$。

6.10 [J]The Sky is the Limit

6.10.1 题意简述

求 x 轴上方 n 个等腰三角形的上轮廓长度（不包括x轴上的长度）。

【数据范围】$n \leq 100$。

6.10.2 算法分析

可以发现上轮廓线右若干个转折点组成，而转折点一定是某个三角形顶点或两个三角形的交点。那么把三角形顶点和两两的交点作为离散事件点（最多 $O(n^2)$ 个），我们只需求出每个离散事件点上的最高高度（可以 $O(n)$ 暴力求），相邻事件点所在高度的连线就是轮廓线一部分。把非 x 轴的部分（即两侧至少一个非 0 ）累加即可，也可全部算入后用类似线段覆盖的方法提前减去。

但此题最坑的是精度，如果用向量求交计算交点，精度死活过不去，必须使用解析式算交点。而且还要去掉不存在的交点，否则线段割成条数过多会导致精度变低。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

6.11 [K]Steam Roller

6.11.1 题意简述

$n \times m$ 的带边权的网格地图上，一辆车从 (r_1, c_1) 到 (r_2, c_2) ，出发、到达、转弯前后的道路上的所需时间要×2，求最短时间。

【数据范围】$n, m \leq 100$。

6.11.2 算法分析

$d[i][j][k][s]$ 表示当前在 (i, j) ，朝向为 k ，这里是否转弯（或出发/达到）时的最小时间。转移需要枚举下次朝向，然后计算相应时间。用 $spfa$ 或 $dijkstra$完成转移。

【时间复杂度】$O((n \times m \times 4)^2)$，【空间复杂度】$O(n \times m \times 4)$。

Chapter 7

ACM/ICPC World Finals 2007 (10/10)

7.1 [A]Consanguine Calculations

7.1.1 题意简述

给出父母和孩子三人中两人的血型，求第三人可能的血型。

【数据范围】血型为ABO血型系统和Rh血型系统。

7.1.2 算法分析

两种血型系统可分开判断，Rh血型直接特判，ABO血型枚举结合的两个基因，模拟血型结合过程。或直接手算打表。

【时间复杂度】$O(1)$，【空间复杂度】$O(1)$。

7.2 [B]Containers

7.2.1 题意简述

按顺序把 n 个字母分堆（新增字母放到任意一堆顶部），要求每堆从上到下不递增，最小化分的堆数。

【数据范围】$n \leq 1000$。

7.2.2 算法分析

贪心。每次从后往前，取最小的字母，然后从最靠前的该字母处继续扩展。

其实就是最长上升子序列的长度。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

7.3 [C]Grand Prix

7.3.1 题意简述

将斜面上的 n 条首尾相连的线段一起旋转，使每条线段的终点高度不低于起点。

最小化旋转的角度，或输出无解。

【数据范围】$n \leq 10000$。

7.3.2 算法分析

由于是往上的斜坡，则每条线段的终点 x 坐标都要不小于起点。（样例给出了角度= 0的trick。）

点旋转的角度相当于线段倾斜角旋转的角度，可算出每条线可以接受的旋转角度，所有区间的交集为合法区间，变为区间覆盖问题，将区间排序后扫一遍。

【时间复杂度】$O(n \log n)$，【空间复杂度】$O(n)$。

7.4 [D]Jacquard Circuits

7.4.1 题意简述

给出一个 n 条边的格点多边形，将所有边缩小尽量大的倍数，使其仍为格点多边形。然后将该图形的边都变为 $1 \sim m$ 倍得到 m 个图形，求这 m 个图形的内部整点数的和。

【数据范围】$n \leq 1000, m \leq 10^6$。

7.4.2 算法分析

先将多边形收缩，即求所有边上点数的 gcd 。然后求出该多边形的面积 A 和边上点数之和 $B = \sum \text{gcd}$ 。

然后运用皮克定理，$Area = Interior + \frac{Border}{2} - 1$

$$\begin{aligned} Interior &= Area - \frac{Border}{2} + 1 \\ &= \sum_{i=1}^{m} i^2 \times A - i \times \frac{B}{2} + 1 \\ &= \sum_{i=1}^{m} i^2 \times A - \sum_{i=1}^{m} i \times \frac{B}{2} + m \\ &= \frac{(2m+1) \times (m+1) \times m}{6} \times A - \frac{m \times (m-1)}{4} \times B + m \end{aligned}$$

注意读入多边形边上有多余点，需要去掉这些点。还有答案要使用 *unsigned long long* 来存。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

7.5 [E]Collecting Luggage

7.5.1 题意简述

给出 n 边形表示传送带，并给出传送带的速度、人的速度，行李在传送带上的初始位置、人的初始位置，求人拿到行李的最短时间。

保证人的速度大于传送带速度。(注意 uva11425 Collecting Luggage EXTREME!!! 上没有这个条件。)

【数据范围】$n \leq 100$。

7.5.2 算法分析

由于人的速度大于传送带，所以如果某个时刻已经拿到行李了，那么后面只要一直追行李，显然也是可以拿到的。所以拿到行李的时间是满足

二分性质的。

二分时间，可以得到行李的位置和人可以走的长度。可 $floyd$ 求出人到达每个顶点的距离，需要判断是否穿越了传送带（即进入内部，判断是否有不在线段端点上的交点，若没有则判断线段是否在多边形内部，以中点为判定点）。同样求出行李到每个点的距离，若人到行李距离$< Time \times V_{人}$，则可行。

【时间复杂度】$O(n^3 + Case \times n^2)$，【空间复杂度】$O(n^2)$。

7.6 [F]Marble Game

7.6.1 题意简述

边长为 n 的面板上有若干小球和洞，以及若干挡板。每次可以使面板倾斜，小球受重力滚向一侧，直到遇到挡板或边界（遇到洞则填进去）。

求最少的操作次数使所有小球都进入对应编号的洞内。

【数据范围】$2 \leq n \leq 4$。

7.6.2 算法分析

总状态数不多，用地图的hash值做下标，直接从起始状态广搜。搜索时模拟四种操作的影响，转移到新状态。

【时间复杂度】O(状态数)，【空间复杂度】O(状态数)。

7.7 [G]Network

7.7.1 题意简述

有 n 条消息被打包为 m 段传送，但每段到达的顺序不一定有序，需要一个缓冲区来重新排序（同一条消息的所有段需要按顺序连续通过缓冲区，可直接通过）。求最优的通过顺序，最小化缓冲区大小。

【数据范围】$n \leq 5, m \leq 1000$。

7.7.2 算法分析

可直接枚举信息通过的顺序。对于当前信息，记录已读取的量，模拟缓冲区。

【时间复杂度】$O(2^n \times m)$，【空间复杂度】$O(m)$。

7.8 [H]Raising the Roof

7.8.1 题意简述

空间中有 n 个点，组成不相交的 V 个三角形。求俯视可见部分的表面积之和。

【数据范围】$n \leq 300, V \leq 1000$。

7.8.2 算法分析

把所有边投影到平面，求出所有交点建平面图，对平面图每个域，该域最上方的面属于同一个三角形，求出该域对应的最上方三角形，通过倾斜角算出对应面积（ 原面积 $\times \frac{1}{\cos\theta}$ ）。

【时间复杂度】$O(V^3)$，【空间复杂度】$O(V^2)$。

7.9 [I]Problem: Water Tanks

7.9.1 题意简述

有 n 个不同高度的水箱，相邻两个之间用高度递增的管道连接。

第一个水箱开口，从第一个水箱开始往内灌水，问最后灌进去的水的体积。

物理知识：封闭空间中气压处处相同，且气压 $\times$ 体积为定值。在一个空气层下的水体，在水面以下D米的水压等于在表面气压加上 $0.097 \times D$ 个大气压。连通水体中同一高度水压相同。

【数据范围】$n \leq 10$。

7.9.2 算法分析

第一个和第二个的连接处以下部分显然倒满。之后每个水箱依次考虑。

当前水箱会有三种状态（该水箱与上一个连接处以下部分水已满，由于连接点递增，前面部分水连通）：

1. 无法到达该水箱与下一个的连接处（判断水达到该高度时的表面水压，和第一个水箱得到的水压比较）：那么设高度为 x，解一元二次方程。

2. 到达下一个连接处，但下一个水箱的高度无法到达连接处的高度（判断方法类似上一个）：该水箱连接处高度的气压为已知值，根据这个计算得到下一个水箱水面高度。

3. 下一个水箱继续做，当前水箱超过连接处部分已封闭，设高度为 x，解一元二次方程。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

7.10 [J]Tunnels

7.10.1 题意简述

在 n 个点 m 条边的图中，要防止一个人从 1 号点到达 0 号点。你可以实时知道他的位置，并在任意时间摧毁图中任意的边。求需要摧毁的最小边数。

【数据范围】$n \leq 50, m \leq 1000$。

7.10.2 算法分析

每点作为源点做最小割，得到每点权值，权值从小到大枚举每点，用阻止他到该点的费用（最小割）+ 这点删掉后的最小割（两部分没有交集）来更新原先的每个点的权值。

【时间复杂度】$O(n \times Maxflow(n, m))$，【空间复杂度】$O(n^2 + m)$。

Chapter 8

ACM/ICPC World Finals 2006 (10/10)

8.1 [A]Low Cost Air Travel

8.1.1 题意简述

有 m 种机票，该飞机依次经过 L_i 个点，只能从起点登机，可以在任意中途点离开。

要完成 Q 次旅行，每次旅行要依次经过 L_j 个点，求每次旅行的最小费用以及方案（保证唯一）。

【数据范围】$Q, m \leq 20$, $L \leq 10$。

8.1.2 算法分析

$f[i][j]$ 表示经过了这次旅行的第 i 个点，且当前在 j 点的最小费用，由于 i 相同时会互相转移，用 $dijkstra$ 完成这些转移，即先找费用最小的点，枚举飞机票以及终点来转移。转移时记录下出发时已经经过的点数和使用的机票，递归输出方案。

【时间复杂度】$O(Qm^2L^3)$，【空间复杂度】$O(mL^2)$。

8.2 [B]Remember the A La Mode!

8.2.1 题意简述

给出 n 种薄饼和 m 种冰淇淋的份数，以及每种薄饼和冰淇淋搭配的收益，求最大的收益。

【数据范围】$n, m \leq 50$。

8.2.2 算法分析

源点向薄饼连费用为0，容量为薄饼份数的边，冰淇淋向汇点连费用为0，容量为冰淇淋份数的边，薄饼和冰淇淋之间连费用为收益，容量无限的边。求最大费用最大流。

【时间复杂度】$O(CostFlow(V = n, E = n^2))$，【空间复杂度】$O(n^2)$。

8.3 [C]Ars Longa

8.3.1 题意简述

给出用 m 根杆子连接 n 个等质量的球的雕塑（在底面上的球已固定），问其能否静止，若静止，是否稳定（某些点施加某些力不移动）。

【数据范围】$n, m \leq 100$。

8.3.2 算法分析

每根杆子对两段的球施加的力设为未知数，每个不在地面的点需要满足合力（包括重力）为 0，根据这些等式的列方程，然后高斯消元求解，有解说明静止。

若静止，给每个点一个随机的力，若仍有解，则稳定。

【时间复杂度】$O(n^2m)$，【空间复杂度】$O(nm)$。

8.4 [D]Bipartite Numbers

8.4.1 题意简述

求比 x 大且为 x 的倍数的最小的二段数。二段数定义为 $m\ s\ n\ t$，即 m 个 s 后面接 n 个 t 组成的数，要求 $0 < s < 10$, $0 \le t < 10$, $s \ne t$, $m > 0$, $n > 0$。

【数据范围】$0 < x < 10^5$。

8.4.2 算法分析

首先预处理 i 个 1 $\mod x$ 的结果 $f(i)$，则在知道 $m\ s\ n\ t$ 后可 $O(1)$ 算出其 $\mod x$ 的值，即 $(f(m+n) \times s + f(n) \times (t-s)) \mod x$。

接下来直接暴力枚举 $n+m$，然后枚举 n, s, t。其中枚举 t 的部分可以优化，根据 x 的个位数，t 取某些值时是肯定不可能整除了，不需要枚举。

但这样仍旧是 TLE，将部分答案打表后发现 x 的个位为 0 时答案较大，而这时 t 一定是 0，但 n 过大是没有必要的，0 的个数的增加只能增加 2 和 5 的因子个数，而 10^5 以内，且个位为 0 的数中因子 2 的个数最多为 14，而 $s = 8$ 时可以抵 3 个 2，所以 0 的个数最多只要 11 个，所以 n 只需要枚举到 11。

加了这个优化后速度快了不少，就能通过数据了。

【时间复杂度】$O((m+n) \times n \times 10^2)$,【空间复杂度】$O(n+m)$。

8.5 [E]Bit Compressor

8.5.1 题意简述

一次压缩指 只要能使 01 串的长度减少，就把最长的连续 n 个 1 替换为 n 的二进制表示。

给出压缩后的长度为 m 的串 S，求原始串长为 L，且其中 1 的个数为 N 的串的个数，只需回答是多解，唯一解，还是无解即可。

【数据范围】$m \le 40, N \le L \le 128000$。

8.5.2 算法分析

由于只需回答多解或唯一解或无解，所以最多只需得到两组解。又因为数据范围较小，可直接搜索，搜到第二组解即可退出。

搜索时记录当前解压位置，解压后的长度和 1 的个数，枚举下一段解压的数后看这样解压是否合法即可。由于解压前后必须是 0 ，所以搜索相当于枚举哪些 0 作为原始数据，所以复杂度是 $O(2^m)$ 。

【时间复杂度】$O(2^m)$，【空间复杂度】$O(m)$ 。

8.6 [F]Building a Clock

8.6.1 题意简述

给出 n 个齿轮的齿数，以及原始转轴的转速，要求通过转轴和齿轮构造出分针和时针的转速。求把转轴数、齿轮数、方案字典序分别作为第一、二、三关键字后最小的方案。

两咬合的齿轮的转速×齿数的值为相反数，在同一转轴上的两齿轮转速相同。分针和时针可共用一部分转轴和齿轮。

【数据范围】$n \leq 6$。

8.6.2 算法分析

先搜索两者的公共部分，然后分别搜索分针和时针的方案。要注意使用全局变量(可选用char数组)来记录当前方案，如果用 *string* 传参数会影响复杂度。

可用估价函数剪枝，若当前转速为正，但不等于目标转速，那至少需要两个齿轮。如果为负，那么可能只需要一次，可预处理某种转速的齿轮是否存在，若根本不可能一个齿轮完成，那就至少需要两个齿轮了。若估价 + 当前齿轮数 $> n$ 则剪枝。

【时间复杂度】$O(n!)$，【空间复杂度】$O(1)$。

8.7 [G]Pilgrimage

8.7.1 题意简述

A 和另一群人旅行， A 负责管理经费，途中会有人数增减，但 A 一直在这其中， A 还在进行了记账，有 4 种条目：

1. IN k，$k \leq 20$ 表示加入 k 个人，新增的人每人需要缴纳这时钱的平均数。
2. OUT k，$k \leq 20$ 表示离开了 k 个人，离开的人每人得到这时钱的平均数。
3. COLLECT k，$k \leq 200$ 表示A向每个人收集了 k 元。
4. PAY k，$k \leq 2000$ 表示 A 支出 k 元。

恰好的是，每种条目后钱仍旧是整数。现给出这个账本中的某一页（即这页开始的钱可以为任意数），求这页开始时的所有可能的人数。

【数据范围】账本中某一页条目数 ≤ 50 。

8.7.2 算法分析

可以发现COLLECT不改变是否整除的情况，可以忽略。

只有在人数改变时才有可能发生不能整除的情况，又由于上一次人数改变后钱一定是当前人数的倍数，所以只需考虑这一段中总支出能否被平分即可。

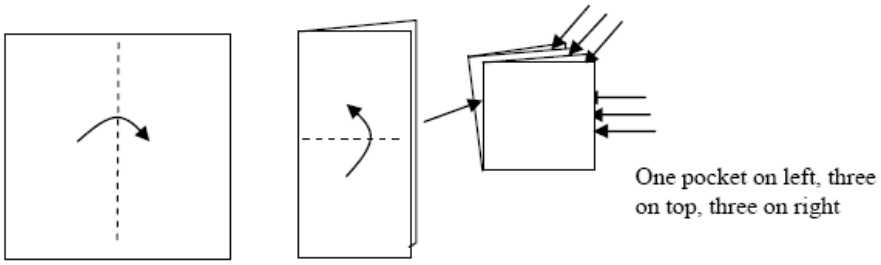
考虑每一段PAY，统计总支出，则当时的人数肯定是该数量的约数，在第一段时统计出所有可能的解，之后逐一筛选。

【时间复杂度】$O(n\sqrt{n \times 2000})$，【空间复杂度】$O(n)$。

8.8 [H]Pocket

8.8.1 题意简述

把一张 $(n+1)\times(n+1)$ 的纸用 k 次沿 n 条平行于坐标轴的折痕的折叠得到 1×1 的形状。问最后形成的口袋的数量。



【数据范围】$n,k\leq 64$。

8.8.2 算法分析

观察最后的纸片，只有在最外面的纸层才对答案有贡献，而且相邻两纸层之间会形成一个口袋。所以口袋数为所有在外面的纸层数量 -4 。

可以逐层展开来计算最后在外面的纸层。先将最后得到的 1×1 的边界标记，然后每次根据操作和之后得到的结果倒推，维护当前图中正面和背面最后会在外面的折痕位置(翻转时，翻转面从正面变为背面，背面变为正面，需要判断是否会超出原范围)。注意边界的翻转对原位置的状态无影响。最后总的标记的折痕数量就是最后可见的折痕数。模拟时可直接用绝对位置来处理翻转，可以通过数据，但存在极端数据使数组存不下，但实际有效位置只有 $O(n^2)$ 个，所以处理成相对位置即可。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

8.9 [I]Degrees of Separation

8.9.1 题意简述

n 个人 m 对关系，一对关系用两个人名表示，求这个图中最短距离最远的两点之间的距离。

【数据范围】$n \leq 50$。

8.9.2 算法分析

先给每个人名一个编号（ map 或 $hash$ 或 $sort$ 判重均可），然后floyd或其他最短路算法求两点之间最短路，取最大的距离。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

8.10 [J]Routing

8.10.1 题意简述

n 个点 m 条边的有向图中，选取最少的中转点，使起点到终点，终点到起点之间都有路径。

【数据范围】$n \leq 100$。

8.10.2 算法分析

显然有一种方案是不管重复点，直接使用两条最短路，但其实利用两条路径中的重合部分可以减少中转点的个数。

而两者公用路径形式为 $A \to C \to D \to B$ 以及 $A \leftarrow D \leftarrow C \leftarrow B$ 。其中有若干段 CD 。

那就可以 DP ，先用 $floyd$ 预处理两点间最短距离，用 $f[i][j]$ 表示当前第一条路径结束点为 i ，第二条路径出发点为 j 的最少中转点个数。若 $i \neq j$ ，新增路径 $i \to j$ ，则可转移到 $f[i][i]$ 或 $f[j][j]$ 或 $f[j][i]$ 。或者新增路径$i \to x$，则转移到 $f[x][j]$ ；新增$j \leftarrow y$，则转移到 $f[i][y]$ 。用 $spfa$ 或 $dijkstra$ 完成转移。

【时间复杂度】$O(spfa(V = n^2, E = n^3))$，【空间复杂度】$O(n^2)$。

Chapter 9

ACM/ICPC World Finals 2005 (10/10)

9.1 [A]Eyeball Benders

9.1.1 题意简述

确定第一张图(n 条线段)是不是第二张图(m 条线段)的某个部分的放大图。所有的线段都被认为是无宽度的。

在第一张图中某条线段的端点在放大前一定是第二张图中某条线段的端点。未放大线段必须距离放大框至少0.005，线段长度至少为0.005。

【数据范围】 $n, m \leq 50$ 。

9.1.2 算法分析

肯定有匹配的端点，则枚举匹配的端点，这样就确定了平移量。然后分两种情况考虑：若第一幅图中关键点在所有线段上，那比例无法确定，但放大比例越大，越容易满足条件，所以直接把放大前长度定为0.005。否则其他线段肯定存在对应线段，枚举某条确定的线段的对应线段，求出比例，然后按比例缩小第一幅图，求出放大框，然后判断是否第一幅图内线段都在第二幅图的放大框内，以及第二幅图的放大框内(包括距离0.005以内

的)的所有线段都在第一幅图内。若两者均成立则可行。

【时间复杂度】$O(nm^2(n+m))$，【空间复杂度】$O(n+m)$。

9.2 [B]Simplified GSM Network

9.2.1 题意简述

B 座塔，平面中的每个点归最近的塔覆盖。有 C 座城市，由 R 条道路连接。有 Q 次旅行，求每次旅行从起点到终点的最少的切换覆盖塔的次数。

【数据范围】$B, C \leq 50$, $R \leq 250$, $Q \leq 10$, 城市不在覆盖区域的边界上。

9.2.2 算法分析

对于每条道路单独计算穿越次数，作为边权。

穿越次数=覆盖塔数−1。枚举一座塔，枚举另一座以不断缩小其区域（只需维护该区域在所求线段上的部分，即一个区间）。若最后仍存在区间，则覆盖到了这条边。

最后 $floyd$ 得到两点间最短路即可。

【时间复杂度】$O(R \times B^2 + C^3)$，【空间复杂度】$O(C^2)$。

9.3 [C]The Traveling Judges Problem

9.3.1 题意简述

n 个点 m 条边的图中，c 个人要去同一个点，多个人经过同一条边只计算一次费用。求最小费用和方案。

最小费用相同时取所有人经过的点集大小最小的方案，也一样时取点集字典序最小的，再次一样时输出任意解。

【数据范围】$n \leq 20$, $c \leq 10$。

9.3.2 算法分析

同一条边费用只算一次即最小生成树。

由于 n 较小，可直接枚举中转点集合，然后直接做最小生成树 $(prim)$ 即可。

由于中转点数量不会达到 n，所以能通过数据。

【时间复杂度】$O(2^{n-c} \times n^2)$，【空间复杂度】$O(n^2)$。

9.4 [D]cNteSahruPfefrlefe(Near Perfect Shuffle)

9.4.1 题意简述

初始为 $[0, 51]$ 的排列，每次操作会取出前 26 个数和后 26 个数，交错地重新组合（最靠前的是原第 26 个数），但最多会有一处错误，交换了相邻两项。

给出最终的排列，问最少进行了几次操作，输出错误所在处。有多个答案时，错误数量少的优先，错误数量一样的按错误的位置组成的数组的字典序小的优先。

【数据范围】$answer \leq 10$。

9.4.2 算法分析

操作次数不同，排列相差很大，直接枚举操作次数，用两数组的差异（可以两两任意交换下的最少的交换次数）当作所需的最少错误次数来估计。

然后搜索，中途用差异估价并剪枝。

【时间复杂度】$O(n^{answer})$，【空间复杂度】$O(n)$。

9.5 [E]Lots of Sunlight

9.5.1 题意简述

给出日出和日落时间和楼房数 n ，各楼房层数 m_i 、房间高度和宽度、楼房间隔，分别求 Q 间房间能被太阳照到的时间段。（只有整段墙都被照到或太阳在楼顶时才算照到。）

【数据范围】$0 \leq n < 100$, $0 \leq m_i < 100$, $Q \leq 1000$, $d_i, w, h \leq 100$。

9.5.2 算法分析

对于每个询问，枚举两侧楼房，用最高的楼层计算挡住的时间（算夹角，用日出日落时间转了 180° 换算成时间），注意不存在该房间的特判。

【时间复杂度】$O(Q \times n)$，【空间复杂度】$O(n)$。

9.6 [F]Crossing Streets

9.6.1 题意简述

给出平面上 n 条与坐标轴垂直的线段，求一个人从起点到终点至少穿过多少条线段（穿过两条重合的线段只算作一条）。他可以任意走，但不能穿过两条垂直的线段的相交处。

【数据范围】$n \leq 500$ ，给出坐标为整数且在 2×10^9 范围内。

9.6.2 算法分析

所有线段会把平面分为若干连通块，相邻连通块之间的距离设为 1 。则起点所在连通块到终点所在连通块的最短路即为答案。

又由于此题 n 范围较小，直接离散出所有坐标。且因为边权都是 0 或 1 ，直接从起点开始 bfs ，用双端队列维护。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n^2)$。

9.7 [G]Tiling the Plane

9.7.1 题意简述

给出一个所有边都和坐标轴平行的 n 边形，问其是否可以铺满平面。
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。

【数据范围】$n \leq 50$。

9.7.2 算法分析

可以发现肯定有一个顶点被选为分割点，则先枚举一个分割点，把边的序列分为两部分，并将一段的边反向且翻转。这样问题就变为能否将序列划分为 2 或 3 段，使其逆序相等，即AB BA或ABC CBA的形式。

而对于每一段，可以发现其中（包括两侧）至少有一个顶点。则先枚举第二段中和第一段的A中第一个顶点匹配的顶点，然后尽量长地往前后扩展，若能扩展到第一段头，第二段尾，则这个A合法。然后问题变为划分为两段，继续这样做下去，直到划完序列或用完 3 段为止。

但这样做是有漏洞了，如果A只有最左侧有顶点，则上述算法枚举不到这种情况，但可以证明这时BC两段的左侧或内部均有顶点，所以翻转整个序列后再进行一次就可以找到了。

【时间复杂度】$O(n^4)$，【空间复杂度】$O(n)$。

9.8 [H]The Great Wall Game

9.8.1 题意简述

$n \times n$ 棋盘上有 n 个棋子，坐标分别为 (x_i, y_i)，一个棋子向四个方向之一移动一格算作一步，求所有棋子排成一行或一列或一对角线所需的最少步数。

【数据范围】。

9.8.2 算法分析

排成一行或一列可以贪心，下面只讨论移到一行的情况，一列的情况类似。

把 x 方向和 y 方向的步数分离，发现 x 方向为和同一数求 abs，所以取中位数最优；而 y 方向为 y_i 和 i 配对，因为配对不交叉，所以最左侧的和第一列配，次左的和第二列配，依次类推，所以排序后直接计算即可。

而排成对角线只能最优匹配，用最小费用流或 KM 计算。

【时间复杂度】$O(n + n^3)$，【空间复杂度】$O(n^2)$。

9.9 [I]Workshops

9.9.1 题意简述

有 n 个会议，每个需要 t_i 时间，有 p_i 的人参与。有 m 个房间，每个可用事件为 t_j ,可以容纳 p_j 个人。给会议安排房间，使可以开会的人最多。

【数据范围】$n, m \leq 1000$。

9.9.2 算法分析

若一个房间 j 可以容纳会议 i，则 j 和 i 可以匹配，价值为人数，直接费用流或 KM 求最优匹配，可能可以卡过。

但此题可以贪心，两者都按时间 t 排序，对于当前的房间在时间允许下显然选人数满足情况下人数最多的会议，用 $multiset$ 维护即可。

【时间复杂度】$O((n+m)\log m + n\log n)$，【空间复杂度】$O(n+m)$。

9.10 [J]Zones

9.10.1 题意简述

给出 n 个每个塔的覆盖人数和 m 个公共覆盖区的人数和涉及的塔。要求只能建造其中 s 座塔，求最多能覆盖的人数。

【数据范围】$n \leq 20$, $m \leq 10$。

9.10.2 算法分析

先把公共覆盖区的人数从塔的覆盖人数中减去，然后搜索选取的集合，记录塔能覆盖的人数，然后枚举每个公共覆盖区，若该区域内存在选取的塔（用二进制优化判断），则累加人数，取最优解即可。

【时间复杂度】$O(2^n \times m)$，【空间复杂度】$O(n+m)$。

Chapter 10

ACM/ICPC World Finals 2004 (10/10)

10.1 [A]Carl the Ant

10.1.1 题意简述

给出第一只蚂蚁的爬行路线（由 n 条平行于坐标轴的线段组成），其他蚂蚁（包括第一只共 m 只蚂蚁）以固定时间间隔出现在洞口，并跟随第一只蚂蚁的路线。当其他蚂蚁到达一个交叉点的时候，它们总是沿着最新的路走。

如果两只蚂蚁在同一个瞬间到达同一点，在第一只蚂蚁的路上走过了较长距离的蚂蚁优先移动；如果两只蚂蚁不是在同一个瞬间到达同一点，等待时间最长的蚂蚁优先移动。

模拟所有蚂蚁的行动，求这些蚂蚁到达终点的次序，以及第一只蚂蚁和最迟的蚂蚁到达的时间。

【数据范围】$n \leq 50, m \leq 100$, 路线上点的坐标的绝对值 ≤ 100。

10.1.2 算法分析

按题意模拟，在每个时刻记录蚂蚁的位置和方向，行进过程中处理路线上的下一格。在下一时刻开始行动时，可以事先确定蚂蚁行动顺序，给路线上每点编号为第一只蚂蚁最迟一次到达该点的时间，那么所在位置的编号大的蚂蚁先走，然后按顺序判断能否向前爬即可，注意在交叉点的判断。

【时间复杂度】$O(\text{路线长} \times m \log m)$，【空间复杂度】$O(\text{路线长} + m)$。

10.2 [B]Heliport

10.2.1 题意简述

在与坐标轴平行的 n 边形内求一个最大的圆。

【数据范围】$n \leq 20$。

10.2.2 算法分析

二分半径，最后的圆卡住多边形的两边 或 边点 或 两点，分别对应于两线、线圆、两圆求交，求所有圆心，判断是否在多边形内并分别和线段求交判断是否可行。

【时间复杂度】$O(n^3 \log)$，【空间复杂度】$O(n)$。

10.3 [C]Image Is Everything

10.3.1 题意简述

给出包含 $n \times n \times n$ 个小方块的立方体空间内的六面视图，若每个小方块的每个面颜色都相同，求最多存在几个小方块。

【数据范围】$n \leq 10$。

10.3.2 算法分析

先把每个面的视图贴到立方体表面，若某个位置的小方块在其中两个面中的颜色不同，则说明这个位置不存在小方块，则把视图往相应的方向推。直到不存在这样的小方块为止。最后剩下的就是最多能存在的小方块个数。

【时间复杂度】$O(n^6)$，【空间复杂度】$O(n^3)$。

10.4 [D]Insecure in Prague

10.4.1 题意简述

有一种加密方法是对于长度为 n 的串 p，选取数字 $m \geq 2n$，以及整数 $0 \leq s, t, i, j < m$ 且满足 $i < j$。长度为 m 的目标串初始所有格为空，然后从位置 s 开始，把 p 的每个字符每隔 i 个空位放入其中。然后再从位置 t 开始，把 p 的每个字符每隔 j 个空位放入其中。最后剩余的位置放上随机的字符。现在给出加密完毕的串，问可能的最长的原串是什么（多解输出多解信息）。

【数据范围】$m \leq 40$。

10.4.2 算法分析

加密后的串长不是很长，所以可以直接枚举原串长度，然后判断是否有解。

需要枚举 n, s, t, i, j，复杂度已达到 $O(m^5)$，而如果暴力判断的话复杂度难以承受，所以需要优化判断部分。

发现在长度确定时，按每隔几个空位放一个得到的位置是确定的，起始位置只是把最后得到的位置循环变换了一下。所以可以按长度 n 和间隔 i 预处理数组 $f[n][i][j]$ 表示第 j 次放下的位置。有了这个数组以后就可以 $O(n)$ 来判断方案是否合法了。

另外如果在得到原串继续判断前统计原串每种字符出现次数，并和剩余串比较并剪枝可以加速。

【时间复杂度】$O(m^6)$，【空间复杂度】$O(m^3)$。

10.5 [E]Intersecting Dates

10.5.1 题意简述

给出 n 段已经覆盖的日期段，和 m 段询问的日期段，求所有询问到未被覆盖的日期段。

【数据范围】$n, m \leq 100$, 年份从 1700 年到 2100 年。

10.5.2 算法分析

首先需要支持把日期转为编号，把标号转为日期输出。较为简便的方法是按顺序枚举所有日期，依次编号，但这样的复杂度是$O(year \times month \times day)$。其实可以记一个按年份的天数前缀和和按月份的天数前缀和（闰年特判）后，就可以方便地把日期转为编号了。但编号转为日期时只能把每年都当 365 天估算，然后微调。每月当 30 天，然后微调（特别注意闰年）。

经过上面的处理，问题变为给出若干区间，找出询问中未被覆盖的区间。由于此题日期范围较小，比较简便的方法是直接用 *bool* 数组记录每个日期的覆盖情况，然后查询，但这样复杂度略高了些。其实可以将覆盖区间和询问区间一起排序，然后用扫描线扫过去，若出现某段未被覆盖且处在询问中则需要输出。然后合并这些区间（或边做边合并）即可。

【时间复杂度】$O(n \log n + m \log m)$，【空间复杂度】$O(n + m)$。

10.6 [F]Merging Maps

10.6.1 题意简述

给出 t 幅地图，大小为 n_i 行， m_i 列（地图每行后可能有**任意**多余字符）。两幅地图合并的分数为所有重合方案中的最高分数。一种重合方案由行偏移量和列偏移量确定，分数为按偏移量重合后重叠部分中相同的字母（'-' 不算）个数，若存在某格的字母（'-'不算）不同则不能重合。要求每次找到当前能得到最高分数的方案进行地图合并（有相同则编号小的优先），合并后的地图编号为下一个编号，直到不能得到更多分数为止。最后输出所有存在的地图。

【数据范围】$t, n_i, m_i \leq 10$。

10.6.2 算法分析

首先枚举两幅图片，枚举偏移量，计算最优值（不重合方案可以把分数设为 $-\inf$，但注意 inf 不能设得太大防止爆 int）。

然后按题意选取合并的地图，合并产生的新地图与现存的其他地图计算最优值。

【时间复杂度】$O(t^2 \times (tn)^2 \times (tm)^2)$，【空间复杂度】$O(t \times tn \times tm)$。

10.7 [G]Navigation

10.7.1 题意简述

给出 n 个卫星在基准时间的位置，运动方向，以及发出信号的时间。一人用导航仪在时刻 t 得到了上述信息，判断他所在的位置（多解、无解、唯一解），若唯一解输出其已到达目标位置或目标位置的角度。

题目假设卫星运动速度均为 $100m/s$，信号速度为 $350m/s$。由于钟的同步误差，若两点距离 $\leq 0.1m$ 视为相同。

【数据范围】$n \leq 10$。

10.7.2 算法分析

每个卫星信号经过处理后变为一个圆的信息，当前位置需要满足所有圆的信息。所以可以随便取两个不同的圆求交（要先判掉不相交[外离、内含]的情况，然后解方程 或 向量法均可），然后对两个交点分别判断是否可行。

由于题目给出距离 ≤ 0.1 视为相同的条件，所以 eps 取 0.1。此外要注意特判 $n = 1$ 的情况。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

10.8 [H]Tree-Lined Streets

10.8.1 题意简述

给出 n 条线段表示街道，街道上每隔 $50m$ 可种一颗树，而距离十字路口 $< 25m$ 处不允许种树，求所有街道上最多能种的树的总和。

【数据范围】$n \leq 100$, 保证街道端点处不是十字路口，没有三线共点，相邻十字路口的距离与 25 的整倍数最小的差至少为 0.001 。

10.8.2 算法分析

由十字路口隔开的每段街道互不影响，所以对线段求交并切割后分别计算即可。要注意的是十字路口对街道的影响是长度 $-25m$ ，然后当成两端均是端点的情况处理。

两端均是端点的街道中最多能种的树的个数为 $\lfloor \frac{len}{50} \rfloor + 1$ 。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n)$。

10.9 [I]Suspense!

10.9.1 题意简述

分别住在两幢楼的 n 层和 m 层的两人从其窗户下端挂出绳子，绳子悬在空中组成了抛物线，他们还在绳子最低点下方 $1m$ 处搭了纸质桥面组成吊桥。但他们要保证两幢楼内的猫无法通过桥面吃到其他楼内的鸟。已知所有猫只能向上跳 $< 0.5m$ 或向下跳 $< 3m$ ，每间房间高度均为 $3m$ ，窗户下端离房间地面 $1m$ 。要求桥面至少比底面高 $1m$ ，至少比他们的窗户下端低 $2m$ 。给出他们楼下所有住户内的宠物情况，以及两幢楼之间的距离 d ，求能悬挂的最长的绳子长度。

【数据范围】$n, m \leq 25$。

10.9.2 算法分析

显然桥面高度越低，绳的长度越长，所以先求最低的可行桥面高度。可以发现最优情况肯定在事件点处，即 $H_i - 0.5, H_i, H_i + 0.5$ 三个位置，从低

到高枚举这些位置，直到找到合法点，然后求得绳的最低点的高度 h 。

然后建立以绳的最低点为原点坐标系，设绳两端分别为 $(x1, y1), (x2, y2)$ ，则 $y1 = 3 * n - 2 - h, y2 = 3 * m - 2 - h$。设二次函数为 $y = ax^2$ ，则

$$\begin{aligned} a{x_1}^2 &= y_1 \\ a{x_2}^2 &= y_2 \\ x_2 - x_1 &= d \\ x_1 < 0 \quad &, \quad x_2 > 0 \end{aligned}$$

解方程得

$$\begin{aligned} x_1 &= -\sqrt{\frac{y_1}{a}} \\ x_2 &= \sqrt{\frac{y_2}{a}} \\ \sqrt{\frac{y_1}{a}} + \sqrt{\frac{y_2}{a}} &= d \\ \sqrt{y_1} + \sqrt{y_2} &= \sqrt{a}d \\ a &= (\frac{\sqrt{y_1} + \sqrt{y_2}}{d})^2 \end{aligned}$$

得到二次函数式子后需要对抛物线长度积分，即

$$\begin{aligned} f(x) &= ax^2 \\ len &= \int_{x_1}^{x_2} \sqrt{1 + f'(x)^2}\, dx \end{aligned}$$

化简

$$\begin{aligned} & \int \sqrt{1 + f'(x)^2}\, dx \\ = \ & \int \sqrt{1 + (2ax)^2}\, dx \\ = \ & 2a \int \sqrt{\tfrac{1}{4a^2} + x^2}\, dx \end{aligned}$$

查表得

$$\int \sqrt{a^2 + x^2}\, dx = \frac{1}{2}x\sqrt{a^2 + x^2} + \frac{1}{2}a^2 \ln \left| x + \sqrt{a^2 + x^2} \right| + C$$

所以

$$\begin{aligned} len &= 2a\left(\frac{x}{2}\sqrt{\frac{1}{4a^2}+x^2}+\frac{1}{8a^2}\ln\left|x+\sqrt{\frac{1}{4a^2}+x^2}\right|\right) \\ &= ax\sqrt{\frac{1}{4a^2}+x^2}+\frac{1}{4a}\ln\left|x+\sqrt{\frac{1}{4a^2}+x^2}\right| \end{aligned}$$

或设$t=2ax$, 则$x=\frac{t}{2a}, dx=d\frac{t}{2a}=\frac{dt}{2a}$

$$\begin{aligned} len &= \frac{1}{2a}\int\sqrt{1+t^2}\,dt \\ &= \frac{1}{2a}\left(\frac{t}{2}\sqrt{1+t^2}+\frac{1}{2}\ln\left|t+\sqrt{1+t^2}\right|\right) \\ &= \frac{1}{2a}\left(ax\sqrt{1+4a^2x^2}+\frac{1}{2}\ln\left|2ax+\sqrt{1+4a^2x^2}\right|\right) \\ &= ax\sqrt{\frac{1}{4a^2}+x^2}+\frac{1}{4a}\ln\left|2ax+\sqrt{1+4a^2x^2}\right| \end{aligned}$$

可以发现和上一个式子只差了$\ln 2a$这个常数，结果是一样的。

最后把 $x1, x2$ 带入式子即可。

【时间复杂度】$O(n+m)$，【空间复杂度】$O(n+m)$。

10.10 [J]Air Traffic Control

10.10.1 题意简述

给定平面上 n 个，这 n 个点按照 y 坐标大的优先级高， y 坐标相同时 x 坐标大的优先级高的顺序排序。

一个控制范围是指的一个圆心 P 和其所能控制的点的个数 val ，控制范围控制的点集是按照到P的距离从小到大，距离相同按照以上优先级选择val个点。控制范围的半径是P 到控制的点的最远距离。控制范围的边界是以P为圆心，r为半径的圆。

给定m个控制范围所能控制的点数和控制范围边界上的两个点。请确定其所控制的点集。若有多个点集满足题意，依照优先级考虑所有点，所某点属于某控制范围而不属于另一个，那么包含该点的控制范围较优。

请对于所有 i，计算出被 i 个控制范围控制的点的个数。

【数据范围】$n \leq 100$, $m \leq 10$。

10.10.2 算法分析

这本来是一道非常简单的模拟题，由于 uva 上的数据错误导致此题通过率极低。

只需要注意到控制范围的边界上一定会有点集中的一个点即可。枚举那个点就可以三点确定控制范围，按照题意求得控制的点集，然后进行简单的统计。

下面给出具体的算法流程

第一步 读入n个点，将其按照优先级排序。

第二步 依次处理m个控制范围。具体方法如下：

(a) 枚举边界上的点 i

(b) 根据点 i 和给定的两个点计算出圆心和半径，求得在该圆内的点集。

(c) 将该点集与目前点集比较，若更优则取代

(d) 将最后得到的点集的点的计数器加 1

第三步 枚举 i，统计有多少个点的计数器为 i，输出。

至于给定三点求外接圆的方法是非常简单的，直接列出圆的方程解之即可。

【时间复杂度】$O(nm \log n)$，【空间复杂度】$O(n)$。

Chapter 11

ACM/ICPC World Finals 2003 (10/10)

11.1 [A]Building Bridges

11.1.1 题意简述

$n \times m$ 网格图上有若干建筑，建筑包含若干个连通的网格（对角相邻也算连通），两网格之间可以建造桥当且仅当两者处在同行或相邻行或同列或相邻列，费用为沿网格边界连接的长度。桥在中间交叉不算相交。

问最少需要的桥的数量，在此前提下使费用最少。如果不能把所有建筑连在一起要输出最少的剩余连通块数。注意如果建造一座桥，输出时应用 $bridge$ 而不是 $bridges$ 。

【数据范围】$n, m \leq 50$。

11.1.2 算法分析

很明显是个最小生成树问题，首先把建筑连起来，然后求出所有可行的边，用 $kruskal$ 求出最小生成树（或不能连成树则求出最少剩余的连通块数）。

如果暴力找到所有可行边，那么边数会达到 $O(n^3)$ 级别，会 TLE，但注意到这是个网格图上的问题，如果两者连线上存在其他建筑的网格，那么这座桥肯定可以被断开的其中一座或两座替代。所以只需找出所有连线上不存在其他建筑的网格的桥即可，这样桥的数量是 $O(n^2)$ 级别的，可以承受。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n^2)$。

11.2 [B]Light Bulbs

11.2.1 题意简述

一行有 n 个灯， n 个开关，每个开关控制所有与其 位置差≤ 1 的灯。求所需按的开关数量最少的一种方案使灯从初始状态变为目标状态，在此前提下使开关状态压缩为 10 进制后最小。

灯的初始和结束状态都用 10 进制压缩后的形式给出，保证最高位至少有一个是 1。

【数据范围】$n \leq \frac{100}{\log 2}$。

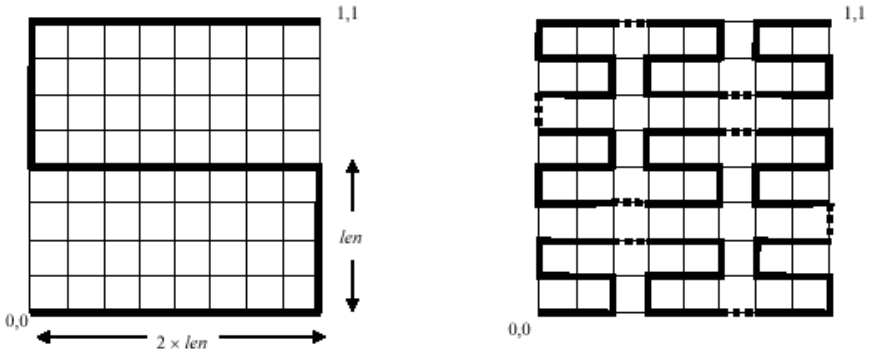
11.2.2 算法分析

注意到 1 号开关的状态确定以后， 1 号灯只能由 2 号开关控制，所以可以推出 2 号开关的状态，之后 2 号灯只能由 3 号开关控制，依次类推可以推出所有开关状态，最后检验最后一个灯的状态是否符合。所以只需枚举 1 号开关的情况后推一遍，两种情况中取最优解。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

11.3 [C]Riding the Bus

11.3.1 题意简述



如上图，两侧分别为 1 阶 2 阶 SZ 曲线。

k 阶 SZ 曲线有 9^k 个连接点，由 9 个 $k-1$ 阶 SZ 曲线组成，由 8 条长度为 len 的线段连接着，连接方法同右图。在 k 阶 SZ 曲线中，$len = \frac{1}{3^k-1}$，所有线段的总长度为 $(9^k-1) \times len$。

给定 SZ 曲线的阶数 n 和起点终点坐标 (x_i, y_i)，求从起点沿直线到最近的 SZ 曲线的连接点，再沿 SZ 曲线到离终点最近的连接点，再沿直线到达终点的距离。如果有多个连接点离一个点最近，则选择 x 坐标最小且 y 坐标最小的一个连接点。

【数据范围】$1 \le n \le 8$, $0 \le x_i, y_i \le 1$。

11.3.2 算法分析

先将图形扩大 3^n-1倍，然后确定最近的关键点，即舍入到最近的整数（0.5 舍去），然后计算出直线走的距离，剩下部分为求两连接点的距离。把两连接点距离变为从左下角分别到两连接点的距离相减后取 abs。

对于某连接点，从高阶到低阶依次确定其在 9 块中的哪一块，需要经过的完整的块中的步数可直接计算，剩余部分需要讨论出口方向，对连接点坐标进行翻转，然后递归该部分，变为原问题。

【时间复杂度】$O(n)$，【空间复杂度】$O(1)$。

11.4 [D]Eurodiffusion

11.4.1 题意简述

n 个国家，每个国家用一个矩形 (xl, yl, xr, yr) 表示，矩形中每个点表示一个城市，矩形不重叠。初始每个城市拥有该国的货币 10^6 个，然后每一天，每个城市都按照它在这一天开始时的余额，将一定量的硬币送给它的所有邻接城市。这“一定量的硬币”指的是对于这个城市目前所拥有的每种图形的硬币，每满 1000 个就要拿出来一个。当某个城市中，每种图形的硬币都至少出现了一个，就称这个城市“已经完成”，当一个国家的所有城市都已经完成的时候，就称这个国家已经完成。你的程序需要得出每个国家的完成时间。

【数据范围】$1 \le n \le 20$, $1 \le xl \le xh \le 10$, $1 \le yl \le yh \le 10$。

11.4.2 算法分析

所有城市完成需要的天数不大，可直接按题意模拟。

对于天数不多可以这样想，大概过1000天可以假设所有货币都从一个城市转移到一个城市，那么经过 1000×最长的最短路长度 天就可以到达所有点。而在 10×10 的地图上最多构造出长度为 59 的链，实测为 189834 天，常数大概为 3 。

【时间复杂度】$O(n \times 1000 \times 10^4)$，【空间复杂度】$O(10^2 + n)$。

11.5 [E]Covering Whole Holes

11.5.1 题意简述

给出两个所有角均为直角的点数分别为 n, m 的多边形表示洞和盖子，问在只能平移的情况下盖子能否完全覆盖洞。

【数据范围】$n, m \le 50$。

11.5.2 算法分析

先把两个多边形都沿横向划分为若干层（沿 x 扫描， y 从下一层加入若干线段的端点，端点的有无性进行异或，最后剩下的数组中，相邻两个组成线段）。

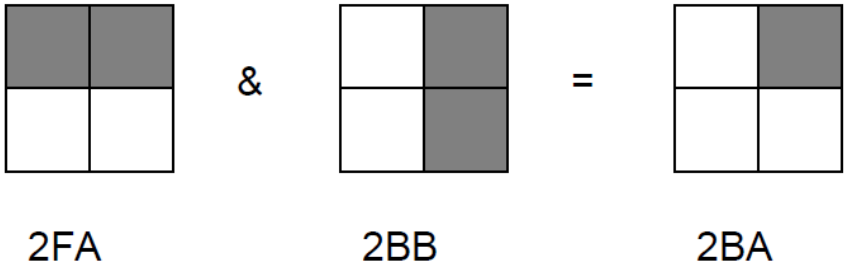
然后枚举两者横向的偏移量（肯定有两边卡住），预处理盖子的某层能否覆盖洞的某层，然后枚举纵向卡住的边，分别往上下统计能否覆盖。

【时间复杂度】$O(n^5)$，【空间复杂度】$O(n^2)$。

11.6 [F]Combining Images

11.6.1 题意简述

一幅长宽相等且为 2 的幂次的黑白图像的四分树压缩是指：若当前树根字符为 1 ，则表示 4 个部分颜色相同，然后给出颜色（ 1 表示黑， 0 表示白）；否则以递归的形式给出四部分的颜色。



给出两幅规模相同的黑白图像的四分树压缩的 16 进制表示，求两者按颜色交的四分树压缩的 16 进制表示。

【数据范围】16 进制表示的编码长度 $n \leq 100$。

11.6.2 算法分析

先转为 2 进制，然后递归比较，若该层某幅图颜色统一，则可直接判断，否则递归四边，注意可能求交完之后四块一样了，要并成一块。最后转回 16 进制。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

11.7 [G]A Linking Loader

11.7.1 题意简述

一组数据包含若干模块，每个模块以 Z 结束。支持以下操作：

1. D symbol offset 定义symbol变量的值为当前内存指针 $+offset$，初始内存指针在 0100_{16} 位置。如果该变量已被定义，则该次定义无效，且标记symbol为重复定义。

2. E symbol 声明变量symbol（每个模块声明的变量分别从 0 开始编号，可能会用到之后定义的变量，但如果该变量一直未定义则值视为 0 ）

3. C n $byte_1$ $byte_2$ ... $byte_n$ 给内存指针开始的 n 个内存赋值，并移动内存指针，其中会引用到该模块中声明的变量的编号，变量的值占用两个内存。

然后输出 16 位校验和 sum，计算方法如下：首先 $sum = 0$，然后按顺序遍历所有地址，每次 sum 循环左移一位（即最高位变为最低位，其他位左移），然后加上该内存地址所存储的值，并保留最后 16 位。

然后按变量名的字典序输出变量名和变量的值（ 16 进制），若该变量未被定义，输出 ???? ，若该变量被重复定义则多输出一个 M 。

【数据范围】D最多 100 个，不存在超过 4 位 16 进制数的内存地址。

11.7.2 算法分析

用 map 对所有出现过的变量重新编号，在内存赋值时对引用变量存变量编号，到最后再处理。按题意模拟即可。

最后输出 map 内所有变量，根据编号得到其值和是否重复定义。

【时间复杂度】$O(16^4)$，【空间复杂度】$O(16^4)$。

11.8 [H]A Spy in the Metro

11.8.1 题意简述

n 个站台的地铁中，有 $m1$ 趟正向列车， $m2$ 趟反向列车，问从 1 号站台出发且 T 时刻到达 n 号站台所需要的最少的在站台上的等待时间（只能在站台换车，可以同一时间从正向列车切换到反向列车，反之亦可。）

【数据范围】$n, m1, m2 \leq 50$, $T \leq 200$。

11.8.2 算法分析

$f[i][j][k]$ 表示时间为 i ，地点为 j ，行进方向为 k （ 0 为在站台， 1 为正向， 2 为逆向）最少等待时间，转移分别讨论继续还是切换行进方向。

【时间复杂度】$O(nT)$，【空间复杂度】$O(nT)$。

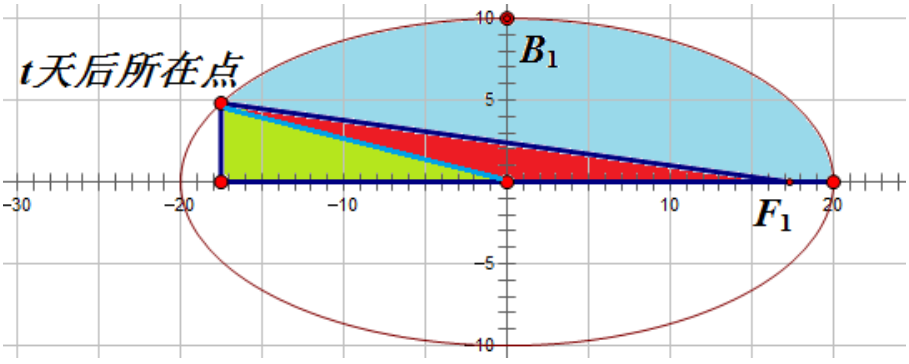
11.9 [I]The Solar System

11.9.1 题意简述

给出第一个行星的轨道的半长轴，半短轴和周期，以及第二个行星的半长轴，半短轴，求第二个行星在第 t 天的坐标。

【数据范围】读入的数均为 int 以内的非负整数，除 t 外均为正整数。

11.9.2 算法分析



如果我们的问题是给出终点，求经过的时间，那么我们只需要算出蓝色区域的面积求和整个椭圆面积的比例即可。计算蓝色区域的面积有两种：

1. **蓝红 – 红**

蓝红区域类似于圆中的扇形，我们可以借助扇形来计算它的面积。椭圆可以看作一个被“压扁”的圆，把椭圆上所有 y 坐标都 $\times\frac{a}{b}$ 后，就变成了半径为 a 的圆，而椭圆中所有图形的面积都$\times\frac{a}{b}$（因为只有 y 变换了，x 保持不变）。所以我们通过坐标变换后可以轻松求出蓝红区域的面积，同时求出椭圆焦点坐标后也可以轻松求出红色区域的面积，这样蓝色区域的面积就搞定了。这种方法比较巧妙。

设 m 为转换为圆后圆心角的弧度， F 为右侧焦点 x 坐标，则

$$t = \frac{m}{2 \times \pi} - \frac{\sin m \times F}{2 \times \pi \times a2}$$

2. **蓝红绿 – 红绿**

蓝红绿就是上半部分椭圆方程下方的面积+下半部分椭圆方程上方的面积，其中第二部分和第一部分的求法相同，我们只讨论第一部分（其实在转化式子后发现可以两部分一起计算）。第一部分的面积可以用积分来求。

首先椭圆方程化为 $y = \frac{b_2}{a_2}\sqrt{{a_2}^2 - x^2}$ （只取 x 轴之上的部分，但其实这样把 x 轴之下部分的积分变正了，两部分可以一起计算了），设 m 为(起点,原点,终点)构成的角的弧度，那么终点的坐标为 $(a_2 \cos m, b_2 \sin m)$。然后按 x 积分，则

$$S = \int_{a_2 \cos m}^{a_2} \frac{b_2}{a_2}\sqrt{{a_2}^2 - x^2}\, dx$$

设 $x = a_2 \cos\theta$ ，则

$$S = \int_m^0 b_2\sqrt{1 - \cos\theta^2}\, da_2 \cos\theta = b_2 \int_m^0 \sin\theta\, da_2 \cos\theta$$

$$\because \frac{da_2 \cos\theta}{d\theta} = -\sin\theta$$

$$\therefore da_2 \cos\theta = -a_2 \sin\theta d\theta$$

$$\therefore S = a_2 b_2 \int_m^0 -\sin^2\theta \, d\theta$$

$$\therefore 2S - a_2 b_2 m = a_2 b_2 \int_m^0 1 - 2\sin^2\theta \, d\theta = a_2 b_2 \int_m^0 \cos 2\theta \, d\theta$$

根据微积分基本定理:

$$\because (\frac{\sin 2\theta}{2})' = \cos 2\theta$$

$$\therefore \int_m^0 \cos 2\theta = \frac{\sin 0}{2} - \frac{\sin 2m}{2} = -\frac{\sin 2m}{2}$$

$$\therefore S = \frac{a_2 b_2 m}{2} - \frac{a_2 b_2 \sin 2m}{4} = \frac{a_2 b_2 m}{2} - \frac{a_2 b_2 \sin m \cos m}{2}$$

红绿是一个直角三角形，可轻松算出面积，即

$$\frac{(F - a_2 \cos m) \times b_2 \sin m}{2} = \frac{b_2 \sin m \times F}{2} - \frac{a_2 b_2 \sin m \cos m}{2}$$

从而求得蓝色部分面积为

$$\frac{a_2 b_2 m}{2} - \frac{b_2 \sin m \times F}{2}$$

而总面积为 $a_2 b_2 \pi$，所以

$$t = \frac{m}{2 \times \pi} - \frac{\sin m \times F}{2 \times \pi \times a2}$$

可以发现求得的式子和上一种方法是一样的。

最后，我们的条件是只知道时间，但随着点逆时针移动，时间单调增，所以我们可以二分点移动的角度，然后判断即可（注意精度问题，eps 需要 10^{-15} 左右）。

同时，如果移动的角度作为未知量的话，就相当于解类似 $ax+b\sin(x)+c=0$ 的方程，这个方程可以用牛顿迭代法解，速度比二分快。

【时间复杂度】$O(\log eps)$，【空间复杂度】$O(1)$。

11.10 [J]Toll

11.10.1 题意简述

从起点运若干物品到终点，中途经过城市时，假设你当前携带的物品数量为 c，则要交出 $\lceil \frac{c}{20} \rceil$ 作为过路费。而经过农村时需要交出 1 个。求初始最少携带的物品数量。

【数据范围】城市用单个大写字母表示，农村用单个小写字母表示，需要运到的物品数量 ≤ 1000。

11.10.2 算法分析

从结束点开始 $dijkstra$ ， $d[i]$表示从该点出发至少需要携带的物品数量，倒推回去，若该点为城市，则用 $\left\lceil \frac{d[i] \times 20}{19} \right\rceil$ 去更新能到达它的点。

【时间复杂度】$O(52^2)$，【空间复杂度】$O(52^2)$。

Chapter 12

ACM/ICPC World Finals 2002 (9/9)

12.1 [A]Balloons in a Box

12.1.1 题意简述

长方体内或外有 n 个备选点，每次可以选一个还未被覆盖且在长方体内的点，在该点放置气球，使其膨胀，直到碰到长方体边界或已有的气球。

最小化长方体内没被气球占据的体积。

【数据范围】$n \leq 6$。

12.1.2 算法分析

搜索放置顺序，计算当前点到已放置气球的点和边界的距离，求最大半径，计算体积。

【时间复杂度】$O(n!)$，【空间复杂度】$O(n)$。

12.2 [B]Undecodable Codes

12.2.1 题意简述

给出一组 n 个二进制编码，找出最短的串（多解找字典序最小的），使其有多种解释方法。

【数据范围】$n \leq 20, L =$ 编码长度 ≤ 20。

12.2.2 算法分析

肯定从开头出现歧义，并使两种情况都有对应解码。任何时候都是在消除另一种情况多余部分，而这个多余部分肯定是某个编码的后缀，用多余部分来表示状态。

类似 $spfa$ 或 $dijkstra$ 求单源最短路，枚举用来消除多余部分的编码来转移。

【时间复杂度】$O(n^2 L \log nL \times L)$，【空间复杂度】$O(nL^2)$。

12.3 [C]Crossing the Desert

12.3.1 题意简述

平面上 n 个点，每走 1 单位距离需要消耗 1 单位水和食物。起点可以购买食物，补充水，中途有若干点可以补充水，不能购买食物但可以储藏食物。且任意时刻能携带的水和食物的上限为 tot 。求从起点走到终点至少需要多少食物。

【数据范围】$n \leq 20$。

12.3.2 算法分析

到终点肯定食物恰好用完，从终点倒推，dijkstra求最短路，根据距离更新其他点到达终点需要储存的最小食物量（先考虑最后一次搬运，然后计算往返搬运的次数）。

需要注意 1 单位食物的输出为 1 unit ，而不是 1 units 。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

12.4 [D]Ferries

12.4.1 题意简述

给出一条线路中每段的属性，若为公路则驾车，速度任意；若为水路则坐渡轮，渡轮在每小时内只有固定的几个时间点才有，且该路程所需时间固定。若驾车限速 $80km/h$，问最早到达时间。并求限速最小为多少，还能以相同时间到达。

【数据范围】$n \leq 10000$(原题未说明 n 的上届)。

12.4.2 算法分析

按 $80km/h$ 的速度，计算每段时间，模拟行驶过程，求出最早时间。然后二分最大速度，用相同的模拟过程，求最早的时间，与之前的时间比较。

【时间复杂度】$O(\log \frac{1}{eps} \times n)$，【空间复杂度】$O(n)$。

12.5 [E]Island Hopping

12.5.1 题意简述

给出 n 个点的位置和权值，两点间距离为欧几里德距离。

选长度和最小的边集使所有点连通，所有边同时开始建，建边的时间正比于边的长度，求每个点和第一个点连通的时间的加权平均数。

【数据范围】$n \leq 50$。

12.5.2 算法分析

长度和最小边集即最小生成树，实际方案对平均值无影响。

在prim时求每个点连到的最短时间。

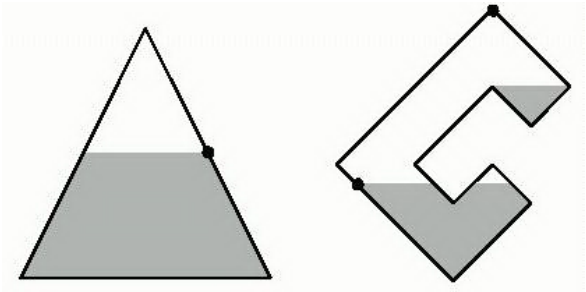
在kruskal时记录每点的总人数。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n^2)$。

12.6 [F]Toil for Oil(WA)

12.6.1 题意简述

给出一个 n 边形，边界上有洞，求该多边形内部能容纳的油的面积，如图中灰色部分。



【数据范围】$n \leq 100$。

12.6.2 算法分析

用平行于所有顶点的直线把多边形切开，那么形成若干梯形，梯形内要么全部有油，要么完全没有。

从下到上扫出每个区域（用当前区间的中位线去切，若多边形的边有交则加入上边界和下边界，最后对两者排序可得到对应关系），若当前连通块下已有洞，那么上面部分就不可能有油，把连通块标记为不可行。而上面部分对当前区间无影响，直接将当前区间内的可行面积累加到答案。

并查集维护连通块，同时用下边界（在当前区间统计前）和上边界（在统计后）上的洞来更新连通块的状态（注意上下边界都要）。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n)$。

12.7 [G]Partitions

12.7.1 题意简述

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。若状态 A 能继续划分得到状态 B，则定义 $A < B$。给出原矩形同为 $w \times h$

的情况 A、 B，求满足 $A < C$ 且 $B < C$ 中最小的 C，以及满足 $D < A$ 且 $D < B$ 中最大的 D。

【数据范围】$w, h \leq 20$。

12.7.2 算法分析

C 就是两种划分的并，直接将所有线求并即可。而 D 是交集，直接求交会产生一些悬空的线，判断悬空的线（存在一侧没有两条垂直边且没有延伸边）并去掉即可。

【时间复杂度】$O(w \times h)$，【空间复杂度】$O(w \times h)$。

12.8 [H]Silly Sort

12.8.1 题意简述

给出 n 个不同的数，每次可以交换任意两个，代价为两数之和。用最小代价将其从小到大排序。

【数据范围】$n \leq 1000$。

12.8.2 算法分析

排序后确定置换群，求出每个置换，对于每个置换有两种方案：

1. 用该置换中的最小值和其他的换。

2. 引入全局最小值，用该最小值和其他的换，然后换出去。

两种方案取最小值即可。

【时间复杂度】$O(n)$，【空间复杂度】$O(n)$。

12.9 [I]Merrily, We Roll Along!

12.9.1 题意简述

给出一条分为 n 段的凹凸不平的道路，求半径为 R 的轮子的圆心的轨迹的长度。

【数据范围】$n \leq 50$。

12.9.2 算法分析

将边扩展成为圆心的轨迹：水平边上移，竖直边向另一侧移，顶点扩展成 90° 圆弧。

求出所有交点（线段和圆弧、圆弧和圆弧），得到圆弧上顺时针下一个，线段正方向下一个交点（开链表记，需要能找到反向边，然后从反向边位置找下一点），走一遍即可找出上轮廓。

也可以边走边求，每次求所有交点，找出最早冲突的，作为这次的旋转角度。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n^2)$。

Chapter 13

ACM/ICPC World Finals 2001 (9/9)

13.1 [A]Airport Configuration

13.1.1 题意简述

求 $2n$ 个门之间的客流量。

【数据范围】$n \leq 25$。

13.1.2 算法分析

按题意模拟、统计。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n^2)$。

13.2 [B]Say Cheese

13.2.1 题意简述

空间中 n 个点，在 r_i 半径的球内可瞬间移动。求起点到终点最短路。

【数据范围】$n \leq 100$。

13.2.2 算法分析

重新计算两两距离， $floyd$ 求两两最短路。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

13.3 [C]Crossword Puzzle

13.3.1 题意简述

给出在 10×10 的地图上进行的填字游戏的 n 个槽(结束点未确定)和 $n+1$ 个备选单词，问哪些可能是多余的。

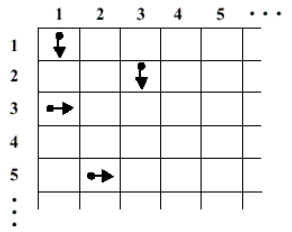


Figure 1: Corner of Example Puzzle

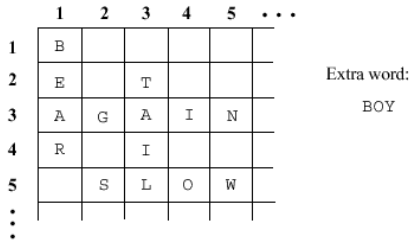


Figure 2: Example Solution

【数据范围】$n \leq 200$。

13.3.2 算法分析

搜索每个槽上放的单词，暴力覆盖。

在搜索时可以先选首字母已确定的槽。

还有若当前集合内的单词都已成为答案，那就没有搜下去的必要了。

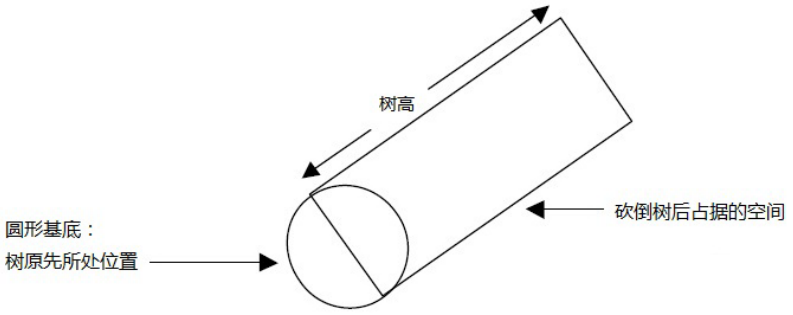
要特别注意槽的前方一格以及单词结尾的后方一格不能填有其他字母。

【时间复杂度】$O(n!)$，【空间复杂度】$O(n)$。

13.4 [D]Can't Cut Down the Forest for the Trees

13.4.1 题意简述

矩形区域内有 n 棵树。一棵半径 r，高度 H 的树被砍倒后会占据 $r \times H$ 的一个矩形区域，可以倒向任意角度。



求最多可以砍倒几棵树，且倒下后不碰到其他树或边界（砍倒后就消失）。

【数据范围】$n \leq 100$。

13.4.2 算法分析

对当前树求出每个限制它的角度。设$L = \sqrt{(r^2 + H^2)}$。树和边界只需考虑半径为 L 的圆。树和树根分三种情况：

1. 若 $H >$ 两圆切线长度，则覆盖区间为切线夹角。
2. 否则若半径为 H 的圆与树根相交，则只需考虑半径为 L 的圆与树根交的弧度。
3. 否则分两个区间，均为半径 L 圆交点~半径 H 圆公切线切点。

然后对角度做区间覆盖，一旦发现可以砍的树则砍倒，更新其他区间。离散区间，类似拓扑排序可做到$O(n^3)$。

【时间复杂度】$O(n^3 logn)$，【空间复杂度】$O(n)$。

13.5 [E]The Geoduck GUI

13.5.1 题意简述

有 k 个物体沿某一矢量在 $n \times m$ 的循环网格上运动(下一次会穿过矢量及其延长线下一个穿过的边线，若穿过交点则先横走)。

要求从中选出两个，使它们经过的格子数之和最多。

注意当物体经过曾经到过的格子时（不论是被它自己还是另一个经过）就会停止。

【数据范围】$n, m \leq 50, k \leq 10$。

13.5.2 算法分析

枚举选取的两个，模拟行走过程，计算经过的格子数。

【时间复杂度】$O(k^2nm)$，【空间复杂度】$O(nm)$。

13.6 [F]A Major Problem

13.6.1 题意简述

音符：C/B# C#/Db D D#/Eb E/Fb F/E# F#/Gb G G#/Ab A A#/Bb B/Cb C/B#

大调音阶：A到G的每个字母将在音阶中出现恰好一次，音阶当中不允许同时出现升调或降调记号。

求把一些音符从一种大调转到另一种大调。

【数据范围】数据组数不超过100组。

13.6.2 算法分析

模拟，记录 12 个半音，判断各种大调的存在情况（2^7枚举），求出 7 个音符，然后利用相对位置找出对应的音符。

【时间复杂度】$O(2^7)$，【空间复杂度】$O(7^2)$。

13.7 [G]Fixed Partition Memory Management

13.7.1 题意简述

n 个内存分区跑 m 个不同的程序需要的时间不同，且内存分区的内存不能超过程序需要的内存。

求最优的分配方案，使所有程序完成的平均时间最小。

【数据范围】$n \leq 10, m \leq 50$。

13.7.2 算法分析

费用流，每个内存分区建m个点，分别表示该内存分区倒数第i次运行，则其对应的时间为原程序运行时间 $\times i$ 。建出所有边，由于同类型边费用递增，所以直接跑最小费用流即可。

【时间复杂度】$O(costflow(nm, nm^2))$，【空间复杂度】$O(nm^2)$。

13.8 [H]Professor Monotonic's Network

13.8.1 题意简述

有 n 个输入和 m 个按顺序的比较器。判断该比较网络是否为排序网络，并求出比较网络所需要的时间。

【数据范围】$n \leq 12, m \leq 150$。

13.8.2 算法分析

根据排序网络01原则，只需枚举 n 个值为 0 或 1 ，然后分别 check 。

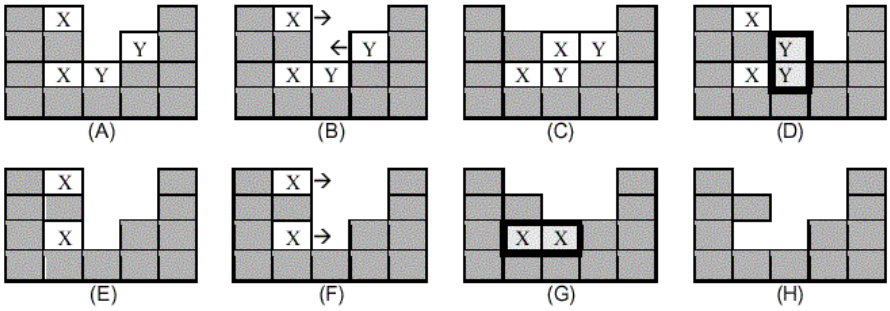
后一问就是求拓扑图的最长路，直接 dp 。

【时间复杂度】$O(2^n \times m)$，【空间复杂度】$O(n)$。

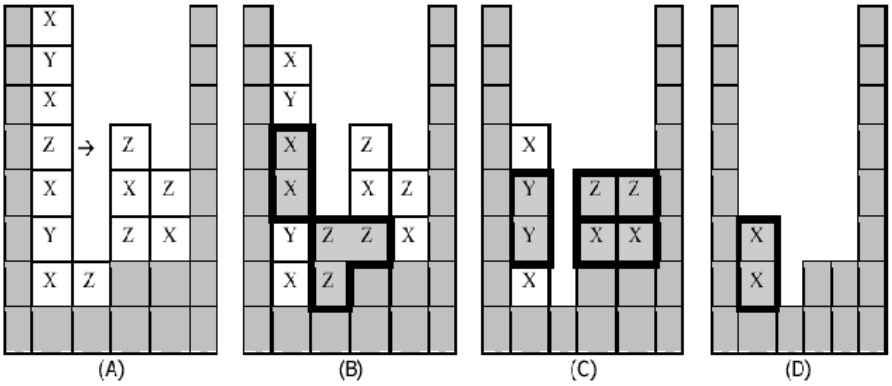
13.9 [I]A Vexing Problem

13.9.1 题意简述

智慧方块的操作为把其中一块向左或右移一格。



消去为至少两个块的连通块，空白部分上方的块会自动掉落，可能引起连锁反映。



给出 $n \times m$ 的智慧方块的局面，求最优解和方案。

【数据范围】$n, m \leq 9$，左右下三侧都是墙，最优解 ≤ 11 步。

13.9.2 算法分析

先对当前状态进行hash，包含当前步数和地图状态，记忆化搜索。然后最优性剪枝，若当前步数 + 估价 $\geq ans$，则剪枝。

我的估价函数为：若某种颜色只剩一个，肯定无解，返回 inf，否则对每种颜色进行DP，忽略方块的行号，相邻两个或三个配对，最少步数为列之间的间隔。

然后对当前决策估价，估价小的优先搜索，估价为刚才的估价函数+局面上所剩的方块数。

【时间复杂度】O(指数级别)，【空间复杂度】O(状态数)。

Chapter 14

ACM/ICPC World Finals 2000 (8/8)

14.1 [A]Abbott's Revenge

14.1.1 题意简述

箭头迷宫，每次走到一个点后会强制向进入处标注的箭头方向之一前进，求最优解。

【数据范围】迷宫大小 $\leq 9 \times 9$。

14.1.2 算法分析

用当前位置和方向来表示状态。直接从起点bfs。

注意终点可能和起点相同。

【时间复杂度】O(迷宫大小 $\times 4^2$)，【空间复杂度】O(迷宫大小)。

14.2 [B]According to Bartjens

14.2.1 题意简述

在一个长度为 n 的数字串中插入 $+ - \times$ 符号（至少一个），使其构成不含前导 0 的算式，且其运算结果为 2000 。

按字典序输出所有解。

【数据范围】$n \leq 9$。

14.2.2 算法分析

直接搜索。

【时间复杂度】O(指数级别)，【空间复杂度】$O(n)$。

14.3 [C]Cutting Chains

14.3.1 题意简述

共 n 个环，要求打开最少数量的环，再闭合，使所有环合并成链状一串。

【数据范围】$n \leq 15$。

14.3.2 算法分析

n 较小，枚举哪些打开，剩下的必须为链，统计链的数量，判断能否用拆开的合并。

【时间复杂度】$O(2^n \times n)$，【空间复杂度】$O(n)$。

14.4 [D]Gifts Large and Small

14.4.1 题意简述

求一个简单 n 边形的面积最小和最大的包围矩形。

【数据范围】$n \leq 100$。

14.4.2 算法分析

先求点集的凸包。

最小：最小矩形覆盖，暴力或旋转卡壳

最大：枚举边界卡住的一点，三分角度，找其余三个卡住的地方。

【时间复杂度】$O(n^2 \log n)$，【空间复杂度】$O(n)$。

14.5 [E]Internet Bandwidth

14.5.1 题意简述

n 个点 m 条边的无向图，求源到汇的最大流。

【数据范围】$n \leq 100, m \leq \frac{n \times (n-1)}{2}$。

14.5.2 算法分析

dinic 或 *sap* 均可。

【时间复杂度】$O(n\sqrt{m})$，【空间复杂度】$O(n+m)$。

14.6 [F]Page Hopping

14.6.1 题意简述

n 个网页之间 m 个单向跳转关系，求两两最短路的平均值。

【数据范围】$n \leq 100, m \leq 100^2$。

14.6.2 算法分析

$floyd$ 或每点一遍 $spfa$ 或 $dijkstra$ 求两点间最短路，求平均。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(n^2)$。

14.7 [G]Queue and A

14.7.1 题意简述

有 n 种任务，每种任务有 b_i 个，模拟工作的分配（按优先级）。

【数据范围】$n \leq 20, b_i \leq 1000$。

14.7.2 算法分析

按题意模拟，在按优先级分配时，可直接按人员优先级排序后按顺序分配工作。

【时间复杂度】$O(n \times b_i)$，【空间复杂度】$O(n)$。

14.8 [H]Stopper Stumper

14.8.1 题意简述

问三个双层同心圆柱（可上下反转）能否平放入一个三棱柱内（高度均相同，同心圆柱都贴住三角形两边）。

【数据范围】无。

14.8.2 算法分析

搜索圆柱和边的对应方式，反转情况（最多一个半径小的向下），然后线段平移求交得到三个圆心，判断。

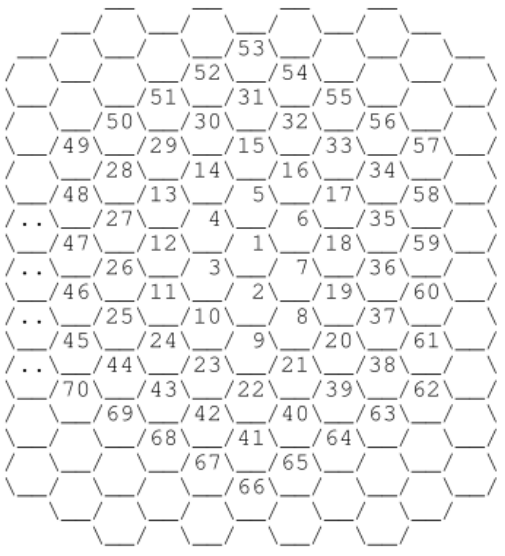
【时间复杂度】$O(1)$，【空间复杂度】$O(1)$。

Chapter 15

ACM/ICPC World Finals 1999 (8/8)

15.1 [A]Bee Breeding

15.1.1 题意简述



用上图方式生成地图并编号，求图中某两点的最短路。

【数据范围】询问点编号 $N \leq 10000$。

15.1.2 算法分析

正六边形坐标转为 60° 角坐标系，按地图生成方式预处理编号对应的位置，直接推式子求距离。

【时间复杂度】$O(N)$，【空间复杂度】$O(N)$。

15.2 [B]Bullet Hole

15.2.1 题意简述

边长为 d 的立方体分为 n^3 块，每块内盛满水，用薄膜隔开。现有一子弹射来，经过的薄膜上会留下弹孔，问最后漏掉了多少水。

【数据范围】$n \leq 50$。

15.2.2 算法分析

每个小方块单独考虑，求其最靠下的弹孔。三维直线和平行于坐标轴的正方体面的交点。即线面求交，判交点是否在正方形内部。（平行可不考虑，和其他面交时可以得到）

但在 uva 的数据中会出现子弹在内部或边界上的情况，因此还必须判断起点是否与面相交。此外，若子弹是往上打的，则需求出整个大正方体的边界和子弹轨迹的交的最小高度，低于这个高度的洞都不会造成漏水。

【时间复杂度】$O(n^3)$，【空间复杂度】$O(1)$。

15.3 [C]A Dicey Problem

15.3.1 题意简述

在 $n \times m$ 的地图上标有数字，只有骰子底面和地图上数字相同或地图上是星星图案时才合法。

求把骰子从起点不回头地滚回到终点需要的最少步数。

【数据范围】$n, m \leq 10$。

15.3.2 算法分析

预处理骰子所有放置情况，模拟骰子的滚动，求出转动后对应的情况。用位置和骰子的放置情况表示状态， bfs 求最优解。

【时间复杂度】$O(n \times m \times 24 \times 4)$，【空间复杂度】$O(n \times n \times 24)$。

15.4 [D]The Fortified Forest

15.4.1 题意简述

给出 n 棵树的位置、权值、高度。砍掉价值最小的树，使其长度能围住剩下的所有树。

【数据范围】$n \leq 15$。

15.4.2 算法分析

枚举每棵树是否砍掉，对剩余求凸包，判断是否可行，注意常数。

【时间复杂度】$O(2^n \times n \log n)$，【空间复杂度】$O(n)$。

15.5 [E]Trade on Verweggistan

15.5.1 题意简述

n个栈，每个中有m_i个数，每个栈中只能取一段前缀，最大化取的数的和。并输出前 10 小的可能的取的个数。

【数据范围】$n \leq 50, m \leq 20$。

15.5.2 算法分析

每个栈求前缀和，找最大值，和最大值一样的均可。背包或暴力维护前 10 小的个数。

【时间复杂度】$O(nm)$，【空间复杂度】$O(n+m)$。

15.6 [F]Robot

15.6.1 题意简述

起点在原点的 n 段机器臂连接，其中奇数段可以在 xOz 平面内旋转，偶数段在 yOz 平面内。给出每段的旋转角度，问是否有部分的纵坐标为负，或两机器臂相交（距离<0.001视为相交）。求最后一段终点的坐标。

【数据范围】$n \leq 10$。

15.6.2 算法分析

一段旋转会对后面的都产生影响，可以从后往前计算，然后分别平移旋转后面每个位置。或化成旋转矩阵，做矩阵乘法。

之后为三维线段求交，先判直线距离，然后判掉两线平行的情况，最后跨立实验。

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n)$。

15.7 [G]The Letter Carrier's Rounds

15.7.1 题意简述

模拟SMTP会话。

命令：HELO myname

解释：发送发件MTA名字。

命令：MAIL FROM:¡sender¿

解释：发送发件者名字。

命令：RCPT TO:¡user¿

解释：给出一位收件者的名字。

命令：DATA

解释：标志正文的传送开始。接下来若干行为邮件正文，以开始第一个字符为句号的一行结束。

命令：QUIT

解释：结束会话。

接下来是收件MTA的返回码。

返回码：221

解释：关闭连接（在QUIT之后）。

返回码：250

解释：行为正常（MAIL FROM或RCPT TO命令所示名字存在，或邮件正文传输结束）。

返回码：354

解释：开始传送邮件（在DATA之后）

返回码：550

解释：行为异常或无此用户（RCPT TO命令之后为不存在用户）

【数据范围】名字为1到15个数字或字母。消息正文每行最多72个字符。总共最多 n 个MTA。总共最多 n 个用户。总共最多 n 条消息。消息正文总共最多 n 行。所有消息涉及的地址不超过 n 个。$n \leq 10^5$。

15.7.2 算法分析

按题意模拟即可。注意用户名字去重。用 *map* 来 *hash* 字符串可简化代码。

【时间复杂度】$O(n \log n)$，【空间复杂度】$O(n)$。

15.8 [H]Flooded!

15.8.1 题意简述

给出 $n \times m$ 地图每格高度，每格大小为 $10m \times 10m$，先该地区积水 $V\ m^3$（总是高度低的先有积水），求水面高度和积水区域百分比。

【数据范围】。

15.8.2 算法分析

从低到高添加区域，在当前基础上计算积水是否会漫到下一高度。注意判断时要加eps。

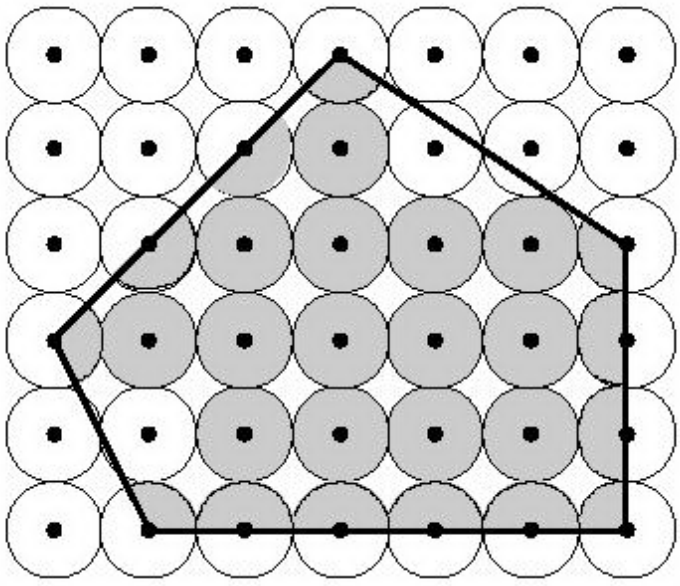
【时间复杂度】$O(nm)$，【空间复杂度】$O(nm)$。

Chapter 16

ACM/ICPC World Finals 1998 (8/8)

16.1 [A]Crystal clear

16.1.1 题意简述



求格点 n 边形内部保持完整的圆或扇形面积之和。

【数据范围】$n \leq 25$，顶点坐标的绝对值 $P \leq 250$。

16.1.2 算法分析

枚举每个圆，判断是否在其多边形内部，然后判断是否被边穿越圆心。只有被边穿过才计数，角穿过不计数，直接算总度数，可以先算内角和，但要注意角上被切割的要减掉。

【时间复杂度】$O(nP^2)$，【空间复杂度】$O(n + P^2)$。

16.2 [B]Flight Planning

16.2.1 题意简述

飞机在不同高度（要满足是某常数的倍数）行驶的油耗由一些常数决定，上升需要油耗。给出路线信息和风速，求最小的油耗总和。

【数据范围】路线数$n \leq 100$。

16.2.2 算法分析

$f[i][j]$ 表示当前第 i 段路线，高度为 j 时最小的耗油总和，枚举下一次的高度转移即可。

【时间复杂度】$O(n \times 20)$，【空间复杂度】$O(n \times 20)$。

16.3 [C]Lead or Gold

16.3.1 题意简述

n 种由三种元素按比例混合得到的若干混合物，问能否用这些按比例混合得到另一种混合物。

【数据范围】$n \leq 100$。

16.3.2 算法分析

比例式化为三者所占分数，则只有两个关键字，若目标状态在其他状态的凸包内，则可行。

注意凸包只有一个点或一条线的情况，注意重复的点。

【时间复杂度】$O(n \log n)$，【空间复杂度】$O(n)$。

16.4 [D]Page Selection by Keyword Matching

16.4.1 题意简述

根据关键字顺序来得出查询和网页的关系强度（对应权值乘积），对每次询问呢输出关系最强的前 5 个网页。

【数据范围】关键字数 $n \leq 8$，网页数 $m \leq 25$。

16.4.2 算法分析

枚举询问，枚举网页，暴力比较，求关系强度，排序。

【时间复杂度】$O(m^2 n)$，【空间复杂度】$O(mn)$。

16.5 [E]Petri Net Simulation

16.5.1 题意简述

有 n 个节点， m 种变换，初始第 i 个节点有 c_i 个令牌。一个变换涉及若干个输入点和输出点，一次变换后所有输入点令牌数 -1（令牌数不能为负），输出点令牌数 $+1$ 。求 T 次变换（或在 T 次变换之前就不能变换了）后的所有点的令牌数。变换可以按任意顺序发生，但保证答案唯一。

【数据范围】$n, m \leq 100$, $c_i \leq 10000$, $T \leq 1000$ ，变换涉及的点数 ≤ 20000。

16.5.2 算法分析

由于保证答案唯一，直接判断每种变换能否发生，若能发生则模拟之即可。

【时间复杂度】$O(T \times n \times m)$，【空间复杂度】$O(n \times m)$。

16.6 [F]Polygon Intersections

16.6.1 题意简述

求边数为 n 和 m 的两个多边形的所有交集。

【数据范围】$n, m \leq 100$。

16.6.2 算法分析

求所有交点，转为平面图，判断每个区域是否在两个多边形内。

在两个多边形边不会重合时，还有更优的做法。交集总是由两多边形的边交替围成，即两个多边形上分别走一步。求出所有交点，从一条内部边开始，走一遍就可以得到所有交集。

【时间复杂度】$O(nm(n+m))$，【空间复杂度】$O(nm)$。

16.7 [G]Spatial Structures

16.7.1 题意简述

用四分树表示来压缩边长为 n 的图像，或把四分图表示转换为图像。

【数据范围】$n \leq 64$， n 是 2 的幂次。

16.7.2 算法分析

按题意模拟

【时间复杂度】$O(n^2)$，【空间复杂度】$O(n^2)$。

16.8 [H]Towers of Powers

16.8.1 题意简述

把 n 分解为 b 进制表示，指数位置上的数需要递归分解。

注意复杂的输出格式，不能有多余空格，每行输出有 80 个字符的长度限制。

【数据范围】 n, b 在无符号 64 位整数范围内。

16.8.2 算法分析

进制分解，模拟输出。

【时间复杂度】$O(log_b n +$ 输出量$)$，【空间复杂度】 $O($输出量$)$ 。