ACM/ICPC World Finals 试题泛做

江苏省常州高级中学 徐毅

January 19, 2014

Contents

1 2013 A Self-Assembly

题目大意

给出 $n(1 \le n \le 40000)$ 种正方形各无数个，且均可以任意旋转或翻转。每种正方形的四条边上都标有“00”、“A+”或“A-”，“A+”和“A-”可以配对，其中 A 可以为任意一种大写字母。能配对的两条边可以相接，要求判断能否用这些正方形连接成无限大的图形，拼接必须在平面上进行。

算法分析

由于正方形可以任意旋转或翻转，我们可以保证每次拼接一定向右或向下，故不存在撞到自身导致无法拼接的问题。因此，假如已知起点边，在拼接的过程中发现又可以接上起点边，那么出现循环即可使图形无限大。

问题转化为给出 $m = 52$ 个点，一个正方形上的每条边对应的点都可以向其他边的匹配边对应的点连边，要求判断图中是否存在环，使用 Floyd 算法即可。

时间复杂度为 $O(m^3 + n)$，空间复杂度为 $O(m^2)$。

2 2013 C Surely You Congest

题目大意

给出一个 $n(1 \le n \le 25000)$ 个点 $m(1 \le m \le 50000)$ 条边的无向图，有 $c(1 \le c \le 1000)$ 个人要从自己所在点出发去点 1，他们只会走最短路，若某时刻两个人同时沿同一方向经过一条边就会发生拥堵。求不发生拥堵的前提下最多有几个人能到达目的地。

算法分析

由于每个人只会走最短路，我们首先用 Dijkstra 算法求出最短路图。

我们发现，两个人如果到点 1 的最短路长度不同，不可能产生拥堵，因此，对于不同的最短路长度我们可以分开处理。

每个子问题就是经典的最大流问题了，用 Dinic 算法来求即可。

时间复杂度为 $O(cmn^2 + (m+n)\log m)$，空间复杂度为 $O(c+m+n)$。

3 2013 D Factors

题目大意

给出一个整数 $n(1 \le n < 2^{63})$，用 $f(k)$ 表示 k 的质因子排列方案数，保证至少有一个 $k(1 \le k < 2^{63})$ 使得 $f(k) = n$，求最小的 k。

算法分析

首先，质因子排列方案数只和每个质因子的次数有关，假如我们知道每个质因子的次数，因为要使 k 最小，必然使最小的质数拥有最大的次数，以此类

推。因此，令 $d = 18$，p_i 表示第 i 小的质数，则最小的 k 必然是

$$\sum_{i=1}^{d} p_i^{q_i}(0 \le q_i \le q_{i-1})$$

，由 DFS 可得其不超过 $w = 50000$ 个。

下面我们只要对每个得到的 k 利用质因数分解求得其 $f(k)$ 并保存在 C++ 的 map 中，对于询问直接输出即可。

时间复杂度为 $O(dw)$，空间复杂度为 $O(w)$。

4 2013 F Low Power

题目大意

有 n 个机器，每个机器有 2 个芯片，每个芯片可以放 k 个电池，芯片能量是 k 个电池能量的最小值。要求将给出的 $2nk(2 \le 2nk \le 10^6)$ 个电池（能量不超过 $c = 10^9$）放在 n 个机器的芯片上，使得所有机器的 2 个芯片能量差的最大值最小。

算法分析

首先我们将 $m = 2nk$ 个芯片按能量从小到大排序，显然每组的两个最小芯片的排位的连线不可能交叉，那么最好的选取策略一定是相邻的两个。

由此，我们二分答案 p，由于第一组是确定的（最小值和次小值），可以往后逐个确定差不超过 p 的组，只要能形成 n 组最小芯片且之后的芯片能将每组芯片补全即合法。

时间复杂度为 $O(m(\log c + \log m))$，空间复杂度为 $O(m)$。

5 2013 H Matryoshka

题目大意

一个完好的套娃集内的玩偶大小是从 1 一直到任意正整数，现在有排成一行的 $n(1 \le n \le 500)$ 个玩偶，每次可以合并相邻的两个套娃集，代价为玩偶数之和减去最小玩偶较小的套娃集中比另一个套娃集最小玩偶小的玩偶数。求将全部玩偶拼成一些完好的套娃集的最小总代价或判断无解。

算法分析

我们首先可以在 $O(n^2)$ 的时间复杂度内预处理出 $s(i,j)$ 表示到玩偶 i 为止大小不超过 j 的玩偶个数和 $m(i,j)$ 表示玩偶 i 到玩偶 j 的最小大小，这样就可以很方便地完成合并代价的计算。

接下来就可以区间动态规划得到 $g(i,j)$ 表示玩偶 i 到玩偶 j 拼成一个完好的套娃集所需的最小代价。

之后再用 $f(i)$ 表示到玩偶 i 为止拼成若干个完好的套娃集所需的最小代价，这个也是比较经典的动态规划。

时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^2)$。

6 2012 A Asteroid Rangers

题目大意

空间中 $n(1 \le n \le 50)$ 个点以固定的速度沿给定的直线运动，每一时刻要用最小的代价使得所有点连通，连边的代价为两点间欧几里得距离。求连边方案的变化次数，保证每一时刻连边方案唯一。

算法分析

每一时刻的连边方案其实就是这个图的最小生成树，由 Kruskal 算法可知发生变化的前提是两条边的大小关系发生变化。显然两点间距离的平方随时间是一个二次函数，因此我们可以通过解方程求出任意两条边大小关系改变的时间。

首先我们用 Prim 算法求出初始时的最小生成树，接下来按照变化时间从小到大依次处理。对于改变大小关系的两条边，由 Kruskal 算法可知两条边同时在或同时不在最小生成树内都不会造成影响，只有不在最小生成树内的那条边 $(x1, y1)$ 变得比在最小生成树内的那条边 $(x2, y2)$ 小，且边 $(x2, y2)$ 在最小生成树内 $x1$ 与 $y1$ 间的路径上时，才会在最小生成树内删去边 $(x2, y2)$ 而加入边 $(x1, y1)$。删边时可以先打个标记，等访问到时再删掉。

时间复杂度为 $O(n^5)$，空间复杂度为 $O(n^4)$。

7 2012 B Curvy Little Bottles

题目大意

瓶子是由一条从 $x = x_l$（瓶底）到 $x = x_h$（瓶口）的 $n(0 \le n \le 10)$ 次多项式曲线绕 x 轴旋转一周构成，从瓶底起体积每增加 $inc(1 \le inc \le 500)$ 要做一个标记。求瓶子的体积及每个标记到瓶底的距离（到第 8 个标记即停止）。

算法分析

令原多项式为 $g(x)$，$f(x) = (g(x))^2$，$h(x)$ 为 $f(x)$ 的原函数，则瓶子体积为

$$\pi \int_{x_l}^{x_h} f(x)\mathrm{d}x$$

，由微积分基本定理可知其为 $\pi(h(x_h) - h(x_l))$。

而标记我们可以依次求，每次二分位置判断体积和 inc 的关系即可。

时间复杂度为 $O(n^2 + n \log c)$，空间复杂度为 $O(n)$。

8 2012 C Bus Tour

题目大意

给出一个 $n(1 \le n \le 20)$ 个点 m 条边且无自环无重边的无向连通图，求起点 1 到终点 n 再回到起点 1 的最短路，要求来回均经过所有点，且去时经过的前 $\lfloor \frac{n}{2} \rfloor$ 个点在回时也是经过的前 $\lfloor \frac{n}{2} \rfloor$ 个点。

算法分析

首先，我们可以用 Floyd 算法预处理出任意两点间最短路。

接下来，我们用 $f(i,j)$ 表示从点 1 出发经过了二进制表示为 i 的点到点 j 的最短路，用 $g(i,j)$ 表示从点 n 出发经过了二进制表示为 i 的点到点 j 的最短路，转移是很简单的。

最后我们只要把两种状态拼接起来就可以得到合法方案，取最小的即可。

时间复杂度为 $O(\binom{n}{\lfloor\frac{n}{2}\rfloor}n^2+2^n n+n^3)$，空间复杂度为 $O(2^n n)$。

9 2012 D Fibonacci Words

题目大意

斐波那契字符串 $f(n)$ 定义为

$$\begin{cases} 0 & n=0 \\ 1 & n=1 \\ f(n-1)+f(n-2) & n\geq 2 \end{cases}$$

，其中 + 为字符串连接。给定一个模式串 $p(1\leq|p|\leq 10^5)$ 和一个数 $n(1\leq n\leq 100)$，求 p 在 $f(n)$ 中的出现次数。

算法分析

易发现 $|f(25)|>10^5$，我们先预处理出 $f(i)(0\leq i\leq 26)$，当 $n\leq 26$ 时我们直接用 KMP 算法得到答案即可。

令 p 在 $f(n)$ 中的出现次数为 $g(n)$，则 $g(n)=g(n-1)+g(n-2)+p$ 在拼接处的出现次数，拼接处为 $f(n-1)$ 的最后 $(p-1)$ 位 $+f(n-2)$ 的最前 $(p-1)$ 位，本质上只有 $f(26)+f(25)$ 和 $f(25)+f(26)$ 两种（根据奇偶），故用 KMP 算法得到 p 在拼接处的出现次数后递推即得答案。

时间复杂度为 $O(n+|p|)$，空间复杂度为 $O(n+|p|)$。

10 2012 E Infiltration

题目大意

给出一个 $n(1\leq n\leq 75)$ 个点的竞赛图，每主动控制一个点还会控制其所有出边指向的点。求需要主动控制最少点数就能控制所有点的一种方案。

算法分析

由竞赛图的性质，所有点的出度和为边数 $\frac{n(n-1)}{2}$。由鸽巢原理的加强形式，至少有一个点的出度为 $\frac{n-1}{2}$。

如果我们每次贪心地取出度最大的点主动控制，再删去所有已被控制的点，那么显然最多只要主动控制 6 个点。

如果有不超过 5 的更优答案，我们可以枚举点集来判断，枚举量为千万级别。

时间复杂度为 $O(\sum_{k=1}^{5}\binom{n}{k})$，空间复杂度为 $O(n^2)$。

11 2012 K Stacking Plates

题目大意

有 $n(1 \leq n \leq 50)$ 堆盘子，第 i 堆有 $h_i(1 \leq h_i \leq 50)$ 个从小到大堆起来的盘子。有两种操作，将一堆盘子一分为二的拆分操作和将一堆堆到另一堆上方（仍满足从小到大）的合并操作。求将这些盘子合并成一堆的最小操作数。

算法分析

令 $m = \sum_{i=1}^{n} h_i$。

首先，在同一堆里每种大小的保留一个即可。要使操作数最少，也就是每堆要尽可能少拆成几个连续部分。

考虑最后合并成的一堆，若某种大小出现了多个，就不可能有某一堆在这里连续放。因此，我们得到如下的贪心策略，根据最后合并成的一堆从小到大依次做，若该种大小只有一个，那么就取唯一的一堆，否则取能延续下去的里面最大盘最小的一堆，取完以后完成该堆的延续。

时间复杂度为 $O(mn)$，空间复杂度为 $O(m)$。

12 2012 L Takeover Wars

题目大意

X 公司有 $n(1 \leq n \leq 10^5)$ 个子公司，Y 公司有 $m(1 \leq m \leq 10^5)$ 个子公司，每个子公司都有其市场价值。公司轮流行动，X 先动，每次可以合并两个子公司（新市场价值为原来两个的和）或干掉对方的一个市场价值比本方的一个子公司小的子公司，先无子公司者败。保证双方之间不会产生子公司市场价值相同的局面，判断 X 是否有必胜策略。

算法分析

首先，假如要合并，一定是合并市场价值最大的两个公司；假如要干，一定是干掉对方市场价值最大的公司。

考虑当前局的决策，若上一局对方选择干，那么本方只能选择合并；若上一局对方选择合并，那么本方能干则干，否则合并。

因此，只要枚举第一局的决策，后面的最优策略都是确定的，只要有一种决策能赢即有必胜策略。

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

13 2011 A To Add or to Multiply

题目大意

$a - C - m(1 \leq a, m \leq 10^9)$ 处理器执行由 A 和 M 指令组成的程序，A 表示数值加 a，M 表示数值乘 m。要保证对于 $[p, q](1 \leq p \leq q \leq 10^9)$ 的输入一定有 $[r, s](1 \leq r \leq s \leq 10^9)$ 的输出，求最短程序（存在多个时要求字典序最小）或判断无解。

算法分析

首先由于数在 10^9 内，显然 M 指令不会超过 30 条，因此我们可以从大到小枚举 M 指令的条数。

接下来的问题就是在其中插入 A 指令，在 n 个 M 指令前插入一个 A 指令相当于在数值上加 am^n，因此插入是独立的，且为了保证程序最短，我们一定会优先插入产生数值大的 A 指令（因为 $am^n = mam^{n-1}$）。

于是，我们只要在输入 q 满足上界的前提下，从高到低插入 A 指令直到输入 p 满足下界。

由于我们从大到小枚举 M 指令条数，当发现当前长度和此前最优长度相等时，当前字典序一定更小。

时间复杂度 $O(1)$，空间复杂度 $O(1)$。

14 2011 C Ancient Messages

题目大意

给出一幅 $n \times m (1 \le n, m \le 200)$ 的黑白图像，按字典序输出图像中出现的如题中图所示的六种象形文字（拓扑结构等价即可），每个文字都是黑色连通块且两两不接触，不存在干扰像素。

算法分析

容易发现，每种文字内部的白色连通块个数分别为 0 到 5，故可以据此判断文字种类。

我们首先 DFS 对所有连通块编号，接下来只要对每个黑色连通块求与其相邻的白色连通块个数，减 1 即为其内部白色连通块个数（当然前提是黑色连通块外侧一圈的白色是连通的，为此我们将原图四周添一圈白色）。

最后统计每种文字的个数，按字典序输出即可。

时间复杂度为 $O(mn)$，空间复杂度为 $O(mn)$。

15 2011 E Coffee Central

题目大意

给出一个 $m \times n (1 \le m, n \le 1000)$ 的网格，其中有 $p (0 \le p \le 5 \times 10^5)$ 个点是咖啡馆。有 $q (1 \le q \le 20)$ 组询问 $d (1 \le d \le 10^6)$，即询问哪个点曼哈顿距离 d 以内的咖啡馆最多。

算法分析

我们把点 (x, y) 变换为 $(x + y, x - y)$，那么就实现了把网格旋转 $45°$，此时距离某点曼哈顿距离为 d 的范围其实是一个边长为 d 的正方形。

那么，我们就可以对存在的咖啡馆位置加 1，枚举每一个点，利用部分和得到其控制范围内的咖啡馆个数，取最多的即可。

时间复杂度为 $O(mnq)$，空间复杂度为 $O(mn)$。

16 2011 H Mining Your Own Business

题目大意

给出一个 $n(1 \le n \le 50000)$ 条边且无重边的无向连通图，要在其中设一些特殊点，使得删掉图中任意一点后每个连通块里至少有一个特殊点。求至少要设特殊点的个数和在此前提下设特殊点的方案数。

算法分析

显然，我们只需要考虑删掉割顶的情况。

我们发现，在删掉所有割顶后形成了很多连通块，只有只与一个割顶相连的连通块会在对应割顶删掉后与其他块分离，因此我们只要对这样的连通块计数即可，方案数即为这些连通块的大小之积。

需要注意的是，当不存在割顶时，仍然需要设两个特殊点（删掉一个时需要另一个）。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

17 2011 I Mummy Madness

题目大意

在沙漠里有 $n(1 \le n \le 10^5)$ 个与人位置不同的木乃伊（坐标绝对值不超过 $c = 10^6$），每一时刻人先走木乃伊再走（8 个方向，可以不走，木乃伊肯定会尽量靠近人）。求最长经过多长时间会被抓住或判断永远不会被抓住。

算法分析

直接求比较困难，考虑二分答案。

那么，人能走到的区域是一个正方形，每个木乃伊能走到的区域也是一个正方形，如果人走到的区域被木乃伊走到的区域的并完全覆盖，则必然被抓住。因此，利用扫描线加线段树求矩形面积并的经典做法就可以判断二分答案的合法性。

接下来就是估计答案的上界。首先，每个时刻木乃伊都可以保证自己和人的距离不会增加（模仿人走），但是如果一个时刻所有木乃伊的最优决策都保持和人的距离不变则人永远不会被抓住。如果人可能被抓住，则每一时刻至少上下左右有一侧的所有木乃伊到人的距离会减少 2，因此被抓住的时间就是木乃伊到人的最远距离级别的。

时间复杂度为 $O(n \log n \log c)$，空间复杂度为 $O(n)$。

18 2011 J Pyramids

题目大意

$\sum_{i=1}^{p} i^2$ 个石块组成底座大小为 $p(p \ge 2)$ 的高金字塔，$\sum_{i=1}^{p} (2i-1)^2$ 个石块组成底座大小为 $2p-1(p \ge 2)$ 的矮金字塔，$\sum_{i=1}^{p} (2i)^2$ 个石块组成底座大小为 $2p(p \ge 2)$ 的矮金字塔。现在有 $n(1 \le n \le 10^6)$ 个石块，求用完所有石块堆成最少个互不相同的金字塔的字典序最大的方案（序指将方案金字塔个数从大到小排列）。

算法分析

我们发现，p 到 150 左右时石块数已超过 10^6，因此设有效的金字塔种数为 m。

接下来我们把金字塔按石块数从小到大排序，发现就是一个 0-1 背包问题，可以用动态规划来解决，方案采用 C++ 的 bitset 来记录即可。

时间复杂度为 $O(mn)$，空间复杂度为 $O(mn)$。

19 2011 K Trash Removal

题目大意

给出一个 $n(1 \le n \le 100)$ 个点的多边形（不自交），要使其通过一竖直管道，求管道的最窄宽度。

算法分析

显然，我们首先要用 Graham 扫描法求出 n 个点的凸包。

那么，我们一定会将凸包的一条边紧靠管道的一边（若斜过来不会更优），此时需要的宽度就是凸包中每个点到这条边距离的最大值。

因此，枚举将哪条边紧靠管道的一边并求出此时的答案，取最小的即可。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$。

20 2010 B Barcodes

题目大意

详见原题。

算法分析

“5% 的误差”的处理是这道题的一大障碍。首先，我们将所有数中最大的一个记为 d，不小于 $\frac{95}{105}d$ 的最小的一个记为 c，最小的一个记为 a，不大于 $\frac{105}{95}a$ 的最大的一个记为 b，则出现的窄区间为 $[a,b]$，宽区间为 $[c,d]$（如有数不在这两个区间内或分隔位不在窄区间内则不合法）。由此，我们可以得到真正的窄宽度属于区间 $[\frac{95}{105}b, \frac{105}{95}a]$，真正的宽宽度属于区间 $[\frac{95}{105}d, \frac{105}{95}c]$，只有真正的窄宽度乘 2 属于的区间与真正的宽宽度属于的区间交集非空才合法。

接下来就是按照题目规则进行判断了，正着和倒着显然只可能存在一种，再分别考虑两个检验字符就可以了。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

21 2010 C Tracking Bio-bots

题目大意

在一个 $m \times n(1 \le m, n \le 10^6)$ 的网格里有 $w(0 \le w \le 1000)$ 堵横向的墙，从一个格子出发只能向右或向上走（不能碰到墙）。求不能到达右上角的格子数。

算法分析

显然横向和纵向的有效坐标个数都是 $2w$，因此通过离散化就可以把网格分成不超过 $4w^2$ 块，之后递推即可得到不能到达右上角的块，将其大小累加即可。

时间复杂度为 $O(w^2)$，空间复杂度为 $O(w^2)$。

22 2010 D Castles

题目大意

给出一棵 $n(1 \le n \le 100)$ 个点的无根树，可以带士兵从任意一个点出发，遍历所有的点，且每条边在每个方向上只能经过一次。当第一次走到某个点时，当前士兵数必须满足该点的最低要求，且会损耗一些士兵，还要留下一些士兵。求一开始要带的最少士兵数。

算法分析

枚举起点作为根就使树转化为有根树。由于每条边在每个方向上只能经过一次，每进入一棵子树就要遍历完该子树的所有结点，因此我们想到使用树形动态规划来解决。

令 $f(i)$ 表示进入以 i 为根的子树所需要的最少士兵数，$s(i)$ 表示进入以 i 为根的子树所消耗的士兵数，易知 $s(i) = \sum_{j \in son_i} s(j)$。

考虑对 $f(i)$ 进行转移，我们需要寻找对儿子的最优遍历顺序。

假设依次遍历 x 和 y，遍历前累计所需人数 u，累计消耗人数 v。

若先遍历 x 后遍历 y，则遍历后累计所需人数 $\max\{u, f(x)+v, f(y)+v+s(x)\}$，累计消耗人数 $v+s(x)+s(y)$。

若先遍历 y 后遍历 x，则遍历后累计所需人数 $\max\{u, f(y)+v, f(x)+v+s(y)\}$，累计消耗人数 $v+s(x)+s(y)$。

我们要找的就是 x 和 y 满足某种条件时有 $\max\{f(x)+v, f(y)+v+s(x)\} \le \max\{f(y)+v, f(x)+v+s(y)\}$。

若 $f(x)+v \ge f(y)+v+s(x)$，必然有 $f(x)+v+s(y) > f(y)+v$，又由 $f(x)+v+s(y) > f(x)+v$，此时必然有 $\max\{f(x)+v, f(y)+v+s(x)\} \le \max\{f(y)+v, f(x)+v+s(y)\}$。

若 $f(y)+v+s(x) \ge f(x)+v$，又由 $f(y)+v+s(x) > f(y)+v$，要使 $f(y)+v+s(x) \le f(x)+v+s(y)$，即 $s(y)+f(y)-s(y)+s(x) \le s(x)+f(x)-s(x)+s(y)$，则必须满足 $f(x)-s(x) \ge f(y)-s(y)$。

由此我们发现，$f(x)-s(x) \ge f(y)-s(y)$ 就是所要找的条件，对 i 的儿子按此从大到小排序进行遍历即可求得 $f(i)$。

时间复杂度为 $O(n^2 \log n)$，空间复杂度为 $O(n)$。

23 2010 G The Islands

题目大意

给出 $n(1 \le n \le 100)$ 个点的坐标（x 坐标单调增）和两个特殊点，要求从点 1 向东经过一些点（包括一个特殊点）到点 n，再从点 n 向西经过剩下的点（包括另一个特殊点）。求最短路径并输出方案。

算法分析

把向西的路径看作向东的另一条路径，由于点的 x 坐标单调增，我们只要依次考虑每个点加入哪条路径，很容易想到使用动态规划来解决这个问题。

令 $f(i, j, 1/0, 1/0)$ 表示第一条路径到点 i，第二条路径到点 j，第一条路径是否已经过特殊点，第二条路径是否已经过特殊点的答案，转移是非常简单的。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

24 2010 I Robots on Ice

题目大意

在一个 $m \times n(1 \leq m, n \leq 8)$ 的网格图上，求从 $(0, 0)$ 出发到 $(0, 1)$ 结束并且在 $\lfloor \frac{imn}{4} \rfloor$ 步到达 $(r_i, c_i)(1 \leq i \leq 3)$ 的路径的条数。

算法分析

DFS，注意到我们的目标是在 4 个规定的步数到达规定的点并且要走完全图，因此可以加两个剪枝，一是当前步数加上当前点到下一个目标点的曼哈顿距离大于要求步数则必然不合法，二是当前点只能往上下两个方向或左右两个方向走，则必然成环（即将原图分成两个连通块），同样不合法。

时间复杂度为 $O(3^{mn})$，空间复杂度为 $O(mn)$。

25 2010 J Sharing Chocolate

题目大意

给出一个长 $x(1 \leq x \leq 100)$ 宽 $y(1 \leq y \leq 100)$ 的巧克力，每次可以沿行或列的分割线将其割开。判断能否把它分成大小指定的 $n(1 \leq n \leq 15)$ 块。

算法分析

我们反过来考虑合并的过程，用 $f(i)$ 表示二进制表示为 i 的块能拼成哪些矩形，通过枚举子集进行转移即可，最后只要判断原矩形是否属于 $f(2^n - 1)$。令可能的矩形总数为 s。

时间复杂度为 $O(3^n + s)$，空间复杂度为 $O(2^n + s)$。

26 2009 A A Careful Approach

题目大意

有 $n(2 \leq n \leq 8)$ 架飞机，第 i 架飞机的着陆时间区间为 $[a_i, b_i](0 \leq a_i \leq b_i \leq 1440)$（以分钟为单位）。求飞机着陆的最小间隔的最大值（以秒为单位）。

算法分析

容易想到二分答案 c，验证工作可以由动态规划完成。令 $f(i)$ 表示已着陆的飞机二进制表示为 i 的最早着陆时间，枚举下一个着陆的飞机 j，若 $\max\{f(i) + c, a_j\} \leq b_j$ 即可进行转移，若 $f(2^n - 1)$ 存在即合法。

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

27 2009 I Struts and Springs

题目大意

平面上有 $n(1 \le n \le 500)$ 个窗户，窗户均与坐标轴平行，且任意两个窗户之间都是包含关系。每个窗户被另一个窗户直接包含，且通过水平三个装置和垂直三个装置（支杆或弹簧，水平和垂直方向上都至少有一个是弹簧）固定于其上。给出最外面窗户的 $m(1 \le m \le 500)$ 次大小变化，求出每次变化后每个窗户的位置和大小（弹簧会等比例缩放且保证为整数）。

算法分析

首先，能够包含一个窗户的最小窗户就是直接包含它的窗户，由此我们可以建出一棵包含关系树并求出其 BFS 序，以得到我们改变窗户的顺序。

将位置和大小转化为三个装置的长度后，改变一个窗户时，我们只要固定支杆，将弹簧等比例缩放即可。

时间复杂度为 $O(n(m+n))$，空间复杂度为 $O(n)$。

28 2008 A Air Conditioning Machinery

题目大意

有 $n=6$ 根相同的 L 型管道（有两个口可用于拼接，如题中图所示），要在长宽高均不超过 $c=20$ 的长方体空间内接通入口和出口，求最少需要的根数或判断 n 根无法实现。

算法分析

由于根据当前接入方向，我们要么接 2 块的一端，要么接 3 块的一端，接上去后可以旋转 4 个方向，故共有 8 种接法，因此我们直接 DFS 就可以了。

时间复杂度为 $O(8^n + c^3)$，空间复杂度为 $O(c^3)$。

29 2008 F Glenbow Museum

题目大意

一个合法的直角多边形内角序列由 R(90°) 和 O(270°) 组成，按逆时针顺序描述，并要求多边形内至少有一点能看到多边形内任意一点。求长度为 $l(1 \le l \le 1000)$ 的合法角序列个数。

算法分析

我们发现，若存在 OO，则不可能存在点看到多边形内任意一点。因此，O 后只能接 R，而加 OR 之后的方向又与加 OR 之前的方向一致，故在序列中加入一个 OR 对序列的合法性没有影响。

由于直角多边形是封闭的，我们必须有且仅有 4 次拐弯，故合法序列为 4 个 R 与若干个 OR 的排列。

由此，我们令 OR 个数 $n=\frac{l-4}{2}$（若 $n<0$ 或 n 不是整数则不存在合法序列），末尾为 R 的合法序列数为 $\frac{(n+4)!}{4!n!}$，末尾为 O 的合法序列数为 $\frac{(n-1+4)!}{4!(n-1)!}$，故答案为 $\frac{(n+1)(n+2)(n+3)(2n+4)}{24}$。

时间复杂度为 $O(1)$，空间复杂度为 $O(1)$。

30 2008 I Password Suspects

题目大意

求长度为 $n(1\le n\le 25)$ 的包含全部 $m(0\le m\le 10)$ 个长度不超过 $l=10$ 的子串的字符串个数，若不超过 42 个按字典序输出方案。

算法分析

首先根据给出的子串建立 AC 自动机，接下来用 $f(i,j,k)$ 表示到第 i 位匹配到 AC 自动机上第 j 个点包含子串的二进制表示为 k 的方案数，递推就可以了。

时间复杂度为 $O(2^m lmn)$，空间复杂度为 $O(2^m lmn)$。

31 2008 J The Sky is the Limit

题目大意

给出 $n(1\le n\le 100)$ 个底边在同一直线上的等腰三角形。求所组成图形的轮廓线长度（底边不算）。

算法分析

显然轮廓线长度为三角形腰的总长减去每条腰在其他三角形内部的长度，为方便处理，首先我们把完全在其他三角形内部且边重合的这些三角形除去。

问题转化为求一端 M 在 $\triangle ABC$ 底边 BC 所在直线上的线段 MN 和 $\triangle ABC$ 的交，那么我们考虑 MN 和 AB,AC 的交点总个数。

若没有交点且 M 在 BC 上，则交为 MN。

若有一个交点 P，当 M 与 B 或 C 重合时，只有 MN 在 $\triangle ABC$ 内部时交为线段 MN；当 M 在 BC 上且不与 B,C 重合时交为 MP；否则交为 NP。

若有两个交点 P,Q 且 A 不与 P,Q 重合，则交为 PQ。

之后对于 MN，把和所有三角形的交求并即可。

时间复杂度为 $O(n^2\log n)$，空间复杂度为 $O(n)$。

32 2008 K Steam Roller

题目大意

给出 $n\times m(1\le n,m\le 100)$ 的网格，有一些格子的边可以走并有权值，若经过一条边前后有转向或出发加速或停止刹车则所花的代价为边权的两倍。求起点到终点的最小总代价。

算法分析

我们需要求到哪个点从哪个方向来到哪个方向去的最短路，也即把一个点拆成 $4 \times 4 = 16$ 个点才能确定边权。

同时还要注意，不能直接把到终点作为结束将所有到终点的边权加倍，因为存在到终点不停之后再绕回来的情况，故我们需另加一个终止点。

接下来用 Dijkstra 算法求最短路即可。

时间复杂度为 $O(mn \log mn)$，空间复杂度为 $O(mn)$。

33 2007 A Consanguine Calculations

题目大意

一个人的血型是 ABO 血型系统和 Rh 血型系统的组合，其受等位基因的控制情况如题中所示。给出父亲、母亲、孩子中两者的血型，判断第三者可能的血型。

算法分析

由于血型情况有限，对于 ABO 血型系统，我们可以对父母的血型打表得到孩子可能的血型，而对于 Rh 血型系统只需要直接判断就可以了。

那么，若给出了父母血型，直接查表即得到孩子血型；若给出父亲（母亲）和孩子的血型，只要找表中父亲（母亲）血型一致、孩子可能是该血型的所有可能的母亲（父亲）血型即可。

时间复杂度为 $O(1)$，空间复杂度为 $O(1)$。

34 2007 G Network

题目大意

有 $n(1 \le n \le 5)$ 条信息，$m(1 \le m \le 1000)$ 个信息包，每个信息的信息包必须按照先后顺序连续通过，不能立即通过的先存储在缓冲区中。给出每个信息包在原信息中的位置及其大小以及所有信息包到达的顺序，求缓冲区大小的最小值。

算法分析

我们首先用 DFS 枚举信息通过的顺序，接下来就是模拟信息包通过的过程了。

时间复杂度为 $O(mn!)$，空间复杂度为 $O(m+n)$。

35 2007 J Tunnels

题目大意

给出一个 $n(1 \le n \le 50)$ 个点 $m(1 \le m \le 1000)$ 条边的无向图，间谍从点 1 出发，你可以在他走的过程中（到达某点时）删掉一些边使得他不能到达点 0。求需要删掉的最少边数。

算法分析

令 $f(i)$ 表示从点 i 出发不能到达点 0 需要删掉的最少边数，则 $f(i) = \min_{j=1}^{n} f(j) + cost(i,j)$，其中 $cost(i,j)$ 表示删掉所有点 $k(f(k) \leq f(j))$ 以后点 i 到点 0 的最小割，可以使用 Dinic 算法求得。

为了确定转移顺序，我们使用 Dijkstra 算法来实现这个过程。

时间复杂度为 $O(mn^4)$，空间复杂度 $O(m+n)$。

36 2006 A Low Cost Air Travel

题目大意

给出 $n(1 \leq n \leq 20)$ 种机票的价格及路线（经过不超过 $c = 10$ 个城市），每种机票可以仅使用从起点到路线中某一个城市为止的一段。现在有 $m(1 \leq m \leq 20)$ 条要走的路线（经过不超过 $c = 10$ 个城市），求每条路线的最小花费及此时顺次使用的机票，保证方案唯一。

算法分析

对于要走每条的路线，我们令 $f(i,j,k)$ 表示已经经过了该路线的前 i 个城市，现在停留在第 j 张机票路线的第 k 的城市的最小花费，那么有继续走当前路线上的下一个城市和从另一张机票路线起点开始走两种转移。

为了确定转移顺序，我们使用 Dijkstra 算法来实现这个过程，令边数 $e = c^2n^2$。

时间复杂度为 $O(em\log e)$，空间复杂度为 $O(e)$。

37 2006 B Remember the A La Mode!

题目大意

给出 $n(1 \leq n \leq 50)$ 种饼片和 $m(1 \leq m \leq 50)$ 种冰淇淋的数量（不超过 100）及两两搭配的收益，两类食物总量相等。求两两搭配的最小（大）总收益。

算法分析

设每类食物总量为 c。将源点向每种饼片连容量为数量费用为 0 的边，将每种冰淇淋向汇点连容量为数量费用为 0 的边，在饼片与冰淇淋之间连容量为正无穷费用为搭配收益的边，用 SPFA 算法求最小（大）费用最大流即可。

时间复杂度为 $O(cmn(m+n))$，空间复杂度为 $O(mn)$。

38 2006 G Pilgrimage

题目大意

给出旅行记录中的一个片段共 $n(1 \leq n \leq 50)$ 条，第 i 条可以是增加 k_i 个人（向增加的每个人收之前平摊到每个人的钱），减少 k_i 个人（给减少的每个人之前平摊到每个人的钱），向每个人收 k_i 元，支出 $k_i(1 \leq k_i \leq 2000)$ 元。要求分钱不出现分数，且整个旅途中至少一直有一个人，判断这段记录开始时可能的人数（无解或列举有限人数或给出无限人数的范围）。

算法分析

设开始时人数为 x，根据加人和减人的操作我们可以得到当前人数 $x+c$。

对于整个旅途中至少有一个人的要求，也就是要满足所有 $x+c>=1$ 的不等式，因此人数的下界是很容易得到的。

对于分钱不出现分数的要求，实际上也就是每两次加人或减人之间支出的钱必须是 $x+c$ 的倍数（收钱操作可忽略，因为已经满足倍数关系；在第一次加减人之前和最后一次加减人之后的支出操作也可忽略，因为那些情况是未知的），可以通过枚举约数来进行不断约束。

若没有有效的支出操作则只有下界，否则解一定是有限个（当然也要满足下界）。

时间复杂度为 $O(kn)$，空间复杂度为 $O(kn)$。

39 2006 I Degrees of Separation

题目大意

给出一个 $n(1 \le n \le 50)$ 个点的无向图，求任意两点间最短路的最大值或判断其不连通。

算法分析

使用 Floyd 算法求任意两点间最短路即可。

时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^2)$。

40 2006 J Routing

题目大意

给出一个 $n(1 \le n \le 100)$ 个点 $m(1 \le m \le 1000)$ 条边的有向图，求从点 1 到点 2 再回到点 1 经过的最少点数（即重复经过一个点只算一次）。

算法分析

我们观察一下路径的情况。首先，在一条路径内一个点必然只经过一次。若两条路径有公共点 A，令两条路径分别为 $1 \to a1 \to A \to a2 \to 2$ 和 $1 \leftarrow b1 \leftarrow A \leftarrow b2 \leftarrow 2$，设 $a1$ 与 $b2$ 仍有公共点 B，那么两条路径一定会走相同的路径 $B \to A$，由此不断分解，我们可以把原来的两条路径中的一个公共段提取出来，之后对于分开的两边再进行同样的分解，可以发现路径一定是公共段与独立段的交错。

由此，我们令 $f(i,j)$ 表示 $1 \to i$ 和 $1 \leftarrow j$ 两条路径所经过的最少点数，假设它们是公共段的终点，则由 $f(j,i)+d(j,i)-1$ 转移过来，$d(j,i)$ 表示从 j 到 i 的最短路，否则从固定一个点，另一个点退一条边的状态转移过来。

$d(i,j)$ 的预处理可以通过从每个点出发 BFS 得到，而转移顺序的确定我们通过 Dijkstra 算法来完成，由于 $f(i,j) \le n$，我们不需要使用堆，直接根据值挂链表即可实现优先队列。

时间复杂度为 $O(mn+n^2)$，空间复杂度为 $O(mn+n^2)$。

41 2005 C The Traveling Judges Problem

题目大意

给出一个 $n(1 \le n \le 20)$ 个点的无向图，保证任意两点间最多只有一条边。给出 $p(1 \le p \le 10)$ 个特殊点，求每个点到点 $dest$ 的一条路径，使得图中路径总长度（即各路径的公共部分只计算一次）最小，保证方案存在。若有多种方案，则要求经过的点数最少。若仍有多种方案，则要求经过的点升序排列后字典序最小。

算法分析

我们发现，最终形成的路径方案一定是一棵以点 $dest$ 为根，以特殊点为叶子结点的树。

如果已知树上有哪些点，只要对这些点求最小生成树就可以了。因此，我们可以先枚举点集，当然特殊点和 $dest$ 必然属于点集，取最小生成树权值最小的即可。

虽然某个最小生成树在以 $dest$ 为根时叶子结点并不一定都是特殊点，但此时一定出现了冗余点，去除冗余点的点集方案一定是更优方案，故最优方案不会被遗漏。

由于给出的可能是完全图，采用 Prim 算法来求最小生成树即可。

时间复杂度为 $O(2^n n^2 + np)$，空间复杂度为 $O(n^2)$。

42 2005 E Lots of Sunlight

题目大意

有 $n(1 \le n \le 100)$ 栋排成一列的楼，每栋楼的层数、楼间距、每间公寓的宽度和高度（所有公寓相同）均是给定的，太阳从日出时间到日落时间以恒定角速度移动。给出 $q(1 \le q \le 1000)$ 组询问，每次询问一间公寓被太阳直射（某一侧被太阳直射或太阳在正上方）的开始时间和结束时间。

算法分析

因为太阳产生的是平行光，显然只有比当前公寓所在层高的楼会对它造成影响。因此，我们只要枚举当前公寓所在楼左侧比它高的楼，算出所需要的阳光与地平面的夹角，取最大值即为开始被直射的夹角，再由日出日落时间推算出此时的时间即为所求的开始时间。同理可再右侧求出结束时间。

时间复杂度为 $O(nq)$，空间复杂度为 $O(n)$。

43 2005 G Tiling the Plane

题目大意

给出一个 $n(1 \le n \le 50)$ 条边（逆时针）周长为 $l(1 \le l \le 50)$ 的直角多边形，判断其能否铺满整个平面。铺满平面只有两种本质不同的情况，棋盘覆盖（A 到 B 与 D 到 C 重合，B 到 C 与 A 到 D 重合）和蜂巢覆盖（A 到 B 与 E 到 D 重合，B 到 C 与 F 到 E 重合，C 到 D 与 A 到 F 重合），其中字母在边界上顺次存在。

算法分析

由于关键点不一定是顶点，我们首先把边都拆成长度为 1 的小段，这样多边形就变为了一个字母序列。

以棋盘覆盖为例，我们枚举 A, B，那么 C 可以直接推得（因为至此必须为一半周长），接下来只要判断 A 逆时针与 E 顺时针的最长公共前缀（这个可以预处理得到）是否不小于 $|AB|$，B 逆时针与 A 顺时针的最长公共前缀是否不小于 $|BC|$ 即可。

时间复杂度为 $O(l^3)$，空间复杂度为 $O(l^2)$。

44 2005 H The Great Wall Game

题目大意

给出 $n \times n (1 \leq n \leq 15)$ 棋盘上 n 个石子的位置，要求将它们移动到构成一行或一列或一条对角线。求最小总步数。

算法分析

我们首先枚举构成哪一行或哪一列或哪条对角线，接下来要确定每个石子移到哪个位置，这其实就是求二分图最优匹配，用 SPFA 算法求最小费用最大流即可。

时间复杂度为 $O(n^5)$，空间复杂度为 $O(n^3)$。

45 2005 I Workshops

题目大意

给出 $n(1 \leq n \leq 1000)$ 个专题讨论会的参加人数与持续时间和 $m(1 \leq m \leq 1000)$ 个房间的容纳人数与开放时间，每个房间只能安排给一个专题讨论会。求不能被安排到房间内的专题讨论会的最少个数和在此基础上的最少人数。

算法分析

我们首先将专题讨论会按持续时间从小到大排序，将房间按开放时间从小到大排序。

比较显然的贪心思想是，按照房间依次处理，将时间上可容纳的专题讨论会加入当前集合，那么该房间一定安排给人数上可容纳的参加人数最多的专题讨论会，整个过程用 C++ 的 multiset 就可以实现了。

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

46 2005 J Zones

题目大意

要在计划的 $n(1 \leq n \leq 20)$ 座服务塔中选 p 座进行建造，每座服务塔能服务一定的人数。此外还有 $m(1 \leq m \leq 10)$ 个公共服务区，描述几座服务塔的公共服务人数。求服务人数最多且在此基础上字典序最小的建塔方案。

算法分析

我们首先枚举要建造的塔，由每座塔的服务人数和公共服务区的服务人数来得到实际服务人数，按题目要求取最优方案即可。

时间复杂度为 $O(\binom{n}{p}mn)$，空间复杂度为 $O(mn)$。

47 2004 D Insecure in Prague

题目大意

一个长度为 n 的大写字母文本的密文长度为 $m(2n \le m \le 40)$，编码方式是选定四个整数 $s,t,i,j(0 \le s,t < m, 0 \le i < j < m)$，一开始有 m 个空位（标记为 0 到 $m-1$），原文首个字符将被放置在第 s 个空位上，接下来每个字符都放置在连续跳 i 个空位后的首个空位上（到末尾则转至开头），结束后从第 t 个位置起的首个空位开始再次输入原文，每次跳 j 个空位，最后在所有未被填充的空位内随机地填入字母。给出密文，求可能被编码的最长原文。

算法分析

首先我们要从大到小枚举 n，接下来还要依次枚举 s,i,t,j，之后的问题就是判断提取出的两段文本是否相同。

若我们在这里暴力模拟跳空位的过程显然时间复杂度是无法承受的，我们发现，第一遍输入原文是在 m 个空位里进行，第二遍输入原文是在 $(m-n)$ 个空位里进行，因此考虑预处理跳空位的过程，用 $f(l,p,k)$ 表示空位个数为 l，初始位置为 0，每次跳 p 个空位，跳 k 次后所处的位置，这样接下来比较两段文本是否相同的过程就可以在 $O(n)$ 的时间复杂度内完成了。

此外还需要加一些剪枝，如利用 s 和 i 得到第一遍输入的原文后，就可根据剩下的字母数判断是否可能完成第二遍输入；选择 t 时显然应该满足第 t 位上是原文的首位字母等等。

时间复杂度为 $O(m^6)$，空间复杂度为 $O(m^3)$。

48 2004 E Intersecting Dates

题目大意

给出 $n(0 \le n \le 100)$ 个已有的日期区间和 $m(0 \le m \le 100)$ 个待查的日期区间（年份在 1700 与 2100 之间），求所有待查的日期区间不在已有的日期区间里的部分。

算法分析

我们预处理所有 $s = 146493$ 个日期对应的编号，对于每个区间 $[l,r]$ 在 l 处 $+1$ 在 $r+1$ 处 -1 并求前缀和即能分别得到已有日期和待查日期的存在部分，所有待查日期存在而已有日期不存在的区间即为答案。

时间复杂度为 $O(m+n+s)$，空间复杂度为 $O(s)$。

49　2004 H Tree-Lined Streets

题目大意

给出 $n(1 \le n \le 100)$ 条线段，现要在线段上安排一些特殊点，同一条线段上的特殊点之间至少隔 50 的距离，特殊点到线段交点至少隔 25 的距离。求最多能安排的特殊点个数。

算法分析

对于每条线段，我们都求出它和其他各线段的交点，这些交点就把这条线段分成了若干小段，对于每个小段分别计算即可。

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

50　2003 A Building Bridges

题目大意

给出一个 $n \times m(1 \le n, m \le 50)$ 的正方形网格，8 连通的黑格为一个建筑，两个建筑可以沿格子的边直线相连，代价为边长。求使所有建筑连通的最小边数以及在此前提下的最小总代价。

算法分析

首先 DFS 得到所有建筑，并处理出建筑之间的所有边，接下来用 Prim 算法求最小生成树即可。

时间复杂度为 $O((mn)^2)$，空间复杂度为 $O((mn)^2)$。

51　2003 B Light Bulbs

题目大意

给出长度不超过 $l(1 \le l \le 100)$ 的两个整数，其转成二进制后分别表示灯泡的起始和目标亮暗情况，开关 i 控制灯泡 $i-1$（若存在）、灯泡 i 和灯泡 $i+1$（若存在），转换开关会使控制的灯泡亮暗取反。求需要转换开关个数最少时转成十进制后最小的转换方案。

算法分析

令灯泡数为 n。

我们枚举开关 1 是否转换，则开关 $i+1$ 是否转换将由灯泡 i 是否达到目标状态来决定，最后灯泡 n 若达到目标状态即得合法解，取转换的开关数少的一个即可。

其中进制转换的过程需要涉及高精度计算。

时间复杂度为 $O(ln)$，空间复杂度为 $O(n)$。

52 2003 F Combining Images

题目大意

给出两张图片的十六进制四分树编码（长度不超过 $l = 100$），求它们的交的十六进制四分树编码。

算法分析

首先将十六进制转为二进制，我们根据两编码的当前位分类讨论进行递归处理。

同为 1 则交的当前位为 1，下一位取决于两编码的下一位是否都为 1。

同为 0 则先假定交的当前位为 0，递归处理 4 遍，注意若 4 遍的结果都是 10 则交的当前位为 1 下一位为 0。

不相同则看当前位为 1 的编码下一位，若为 1 则将当前位为 0 的编码当前段复制给交，若为 0 则将当前位为 0 的编码当前段跳过，且交的当前位为 1 下一位为 0。

时间复杂度为 $O(l)$，空间复杂度为 $O(l)$。

53 2003 H A Spy in the Metro

题目大意

地铁有 $n(1 \leq n \leq 50)$ 站，列车双向运行，在相邻站间所需的运行时间是给定的，列车停靠和人换乘都是瞬间完成。给出 $m1(1 \leq m1 \leq 50)$ 辆从首站出发的列车的发车时间和 $m2(1 \leq m2 \leq 50)$ 辆从末站出发的列车的发车时间，求时刻 0 从首站出发并于时刻 $t(0 \leq t \leq 200)$ 到达末站在车站等待的最少总时间或判断无法到达。

算法分析

容易想到用动态规划来解决这个问题。令 $f(i, j)$ 表示时刻 i 到达站 j 在车站等待的最少总时间，一种转移是继续停留，另一种转移是若此刻有车在站 j 则乘车，至于判断此刻是否有车通过预处理就可以完成了。

时间复杂度为 $O(n(m1 + m2 + t))$，空间复杂度为 $O(nt)$。

54 2003 J Toll

题目大意

给出一个 $n = 52$ 个点的无向图，要将 $p(1 \leq p \leq 1000)$ 个货物从点 x 运到点 y，进入一个大写字母点需要付出 $\lceil \frac{cur}{20} \rceil$ 个货物的代价（cur 表示当前货物量），进入一个小写字母点需要付出 1 个货物的代价。求最少要带的货物量。

算法分析

令 $f(i)$ 表示从点 i 到点 y 需要的最小货物量，显然我们应该以 $f(y) = p$ 作为初始状态开始求，可以想到借助 Dijkstra 算法。

所不同的是，边 $< i, j >$ 的权值不是事先知道的，而是在得到 $f(i)$ 以后，若 i 为大写字母，则通过二分来更新 $f(j)$，否则通过 $f(i) + 1$ 来更新 $f(j)$。

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

55 2002 A Ballons in a Box

题目大意

给出一个长方体盒子的顶点坐标和 $n(1 \leq n \leq 6)$ 个点，每个点若在盒子内且不在其他气球内即可放置气球，放置的过程为以该点为球心膨胀至触及盒子边缘或其他气球，摆放顺序任意。求气球占据的最大体积。

算法分析

用 DFS 枚举摆放顺序，模拟摆放过程，取体积最大的即可。

时间复杂度为 $O(n!)$，空间复杂度为 $O(n)$。

56 2002 C Crossing the Desert

题目大意

给出 $n(1 \leq n \leq 20)$ 个点的坐标和携带单位上限，要从点 1 走到点 n，每走 1 单位要消耗 1 单位水和 1 单位食物，身上携带的水和食物总量不能超过上限。在点 1 可以购买食物，在所有点都可以储存食物和补充无限多的水。求最小要购买的食物量。

算法分析

令 $f(i)$ 表示从点 i 到点 n 需要的最小食物量，显然我们应该以 $f(n) = 0$ 作为初始状态开始求，可以想到借助 Dijkstra 算法。

所不同的是，边 $< i, j >$ 的权值不是事先知道的，而是在得到 $f(i)$ 以后计算得到的。边权实际上也就是运输食物过程中消耗的食物量，计算时要注意，当不能一次运输完时，除最后一次外的每次运输都要考虑来回消耗的食物总量。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

57 2002 E Island Hopping

题目大意

给出 $n(1 \leq n \leq 50)$ 个点的坐标与权值，两点间连边代价为其距离，要求用最小总代价使所有点与点 1 连通。求此时每个点到点 1 路径上最长边乘点权的平均值。

算法分析

我们首先用 Prim 算法求最小生成树，再在树上从点 1 出发 DFS 求得每个点到点 1 的最长边，计算答案即可。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

58 2002 H Silly Sort

题目大意

给出一个长度为 $n(1 \le n \le 1000)$ 的序列，每个数都是不超过 1000 的正整数且互不相同，交换两个数的代价为它们的和。求最小总代价将其升序排序。

算法分析

把每个位置向其排序后的位置连边，就可以得到一些环，每个环在排序过程中都是独立的，因此我们分别处理每个环。

对于一个大小为 k 的环，若只在环内交换，显然将最小的数交换 $(k-1)$ 次，其余各数交换 1 次即可；若引入整个序列的最小值 x，我们先将 x 与环中最小值交换，重复之前的计算后再把 x 换出来。每个环两种方式取最小的累加到答案中即可。

时间复杂度为 $O(n \log n)$，空间复杂度为 $O(n)$。

59 2001 A Airport Configuration

题目大意

给出 $n(1 \le n \le 25)$ 个城市的每个城市要去其他各个城市的人数和 $m(1 \le m \le 20)$ 种城市位置的方案，一个出发城市到一个目标城市距离为其位置差加 1，一种方案对应的客流指数为每个人要走的距离和。将方案按第一关键字客流指数第二关键字编号从小到大排序后输出。

算法分析

按题意进行计算和排序即可。

时间复杂度为 $O(mn^2 + m \log m)$，空间复杂度为 $O(m + n^2)$。

60 2001 B Say Cheese

题目大意

给出 $n(0 \le n \le 100)$ 个球的球心和半径 r_i 以及起点与终点的坐标，在球内穿梭不需要时间，在球外穿梭需要的时间等于经过的距离。求起点到终点的最短时间。

算法分析

把起点与终点看作两个半径为 0 的球，令 $dist(a,b)$ 为球 a 与球 b 的球心距，那么显然从球 a 到球 b 的时间为 $\max\{dist(a,b) - r_a - r_b, 0\}$。

那么，问题转化为求 $n+2$ 个点的无向图上给定两点之间的最短路，使用 Dijkstra 算法即可。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

61 2001 E The Geoduck GUI

题目大意

给出一个 $m \times n (1 \le m, n \le 50)$ 的网格，$k(2 \le k \le 10)$ 只象拔蚌通过每秒种横向或纵向移动一格来模拟不同的矢量（出界则移至相反的一边，若矢量线经过格子的角则先横着走再竖着走），其初始位置为矢量相对网格是向内的一角。所有象拔蚌会同时移动，若一只象拔蚌走进之前被自己或其他象拔蚌访问过的区域或两只象拔蚌同时走进同一个格子或试图交换位置均会停下。要求选出两只象拔蚌使得访问尽量多的格子且在此基础上时间尽量少。

算法分析

首先枚举选择哪两只象拔蚌，接下来就是要用模拟来得到访问的格子数及时间。

以向右上方的矢量为例，在实现过程中唯一的问题就是判断象拔蚌是向右还是向上走，这通过判断矢量所在直线与当前格右边界所在直线的交点是否在当前格上方即可。

时间复杂度为 $O(k^2mn)$，空间复杂度为 $O(k + mn)$。

62 2001 F A Major Problem

题目大意

给出源曲调大调音阶的起始音符和目标曲调大调音阶的起始音符，判断这两个大调音阶是否存在。若存在，还将给出一些音符，若在源曲调大调音阶里出现，求目标曲调大调音阶对应位置的音符。

算法分析

令大调音阶音符数 $n = 8$，按大调音阶的规则模拟生成过程即可，借助 C++ 的 map 完成字符串与编号的映射。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

63 2001 G Fixed Partition Memory Managemen-t

题目大意

有 $m(1 \le m \le 10)$ 个内存分区和 $n(1 \le n \le 50)$ 个正等待运行的程序，每个程序在不同大小内存分区的运行时间不同。求一种合法分配方案（即没有两个程序同一时间在同一个内存分区内被运行）使得所有程序运行结束时间的和最小。

算法分析

一个程序对时间和的贡献值取决于内存分区和它是该内存分区倒数第几个运行的，于是我们想到把一个内存分区拆成 n 个点，表示一个内存分区倒数第几个运行程序产生的贡献值。

那么，我们只需要将源点向程序连容量为 1 费用为 0 的边，将各个拆出的内存分区点向汇点连容量为 1 费用为 0 的边，将程序向每个内存分区的每个运行位次连容量为 1 费用相应贡献值的边，那么用 SPFA 算法求出最小费用最大流一定对应着最优且合法的分配方案。

时间复杂度为 $O(m^2n^4)$，空间复杂度为 $O(m^2n^3)$。

64 2001 H Professor Monotonic's Network

题目大意

给出一个 $n(1 \le n \le 12)$ 个比较值 $k(1 \le k \le 150)$ 个比较器的比较网络，判断其是否为排序网络并求出完成所有比较所需的时间。

算法分析

由于不相关的比较可以并行计算，因此每次比较都是在两个比较值上一次比较结束后进行（即时间为两者最大值加 1）。

至于判断是否为排序网络，根据排序网络的 0-1 原理，如果对于属于集合 $\{0,1\}$ 的每个输入值，排序网络都能正确运行，则对任意的输入值，它也能正确运行。因此，只要枚举输入并模拟网络的运行，若全都合法则为排序网络。

时间复杂度为 $O(2^n(n+k))$，空间复杂度为 $O(n+k)$。

65 2000 A Abbott's Revenge

题目大意

给出一个 9×9 的箭头迷宫，对于每个格子，若从某个方向到达它，只能从它对这种方向规定的几种方向出去。求给定起点到给定终点的最短路径（起始方向给定）。

算法分析

用 BFS 求出从每个方向到达每个点的最短路径即可，令总状态数 $s = 9 \times 9 \times 4 = 324$。

时间复杂度为 $O(s)$，空间复杂度为 $O(s)$。

66 2000 B According to Bartjens

题目大意

给出 $n(1 \le n \le 9)$ 个数字，要求在其中插入加减乘号（至少要插一个），使得运算结果为 2000，且分出的数不能有前导零。按字典序输出所有合法方案。

算法分析

用 DFS 枚举每个数字后插入的运算符或是不插入（按乘加减无的顺序即满足字典序要求），最后利用栈计算表达式的值，判断是否为 2000 即可。

时间复杂度为 $O(4^nn)$，空间复杂度为 $O(n)$。

67 2000 C Cutting Chains

题目大意

给出 $n(1 \le n \le 15)$ 个环和一些环的相扣关系，将一个环打开然后关上算作一次操作。求最少操作次数使得环扣成一条链。

算法分析

显然一个环只会被操作一次。枚举每个环是否被操作，首先消去被操作的环，判断剩下的环是否是若干条链（从每个度为 1 的点出发走一遍即可），若是且链数最多比消去的环数多一条即合法，可以更新答案。

时间复杂度为 $O(2^n n^2)$，空间复杂度为 $O(n^2)$。

68 2000 E Internet Bandwidth

题目大意

给出一个 $n(1 \le n \le 100)$ 个点 $m(1 \le m \le \frac{n(n-1)}{2})$ 条边的无向图，每条边有容量。求点 s 到点 t 的最大流。

算法分析

使用 Dinic 算法求最大流即可。

时间复杂度为 $O(mn^2)$，空间复杂度为 $O(m+n)$。

69 2000 F Page Hopping

题目大意

给出一个 $n(1 \le n \le 100)$ 个点的有向图，保证每个点都能到达其他点。求任意两点间的平均最短路。

算法分析

使用 Floyd 算法求任意两点间最短路，求和除以总路径数 $n(n-1)$ 即可。

时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^2)$。

70 2000 G Queue and A

题目大意

有 $n(1 \le n \le 20)$ 个主题的请求，第 i 个主题有请求数 $b_i(1 \le b_i \le 10000)$、第一个请求到达时间、每个请求的处理时间和相邻请求到达的间隔时间。有 $p(1 \le p \le 5)$ 个人来处理这些请求，第 i 个人有 $g_i(1 \le g_i \le n)$ 个可以处理的主题，主题列表按优先级从高到低的顺序给出。一个人空闲时会从自己的主题列表里按优先级从高到低选择一个已到达的请求开始处理，上次开始处理时间早的人优先选择，上次开始处理时间相同则编号小的人优先选择。求处理完所有请求的总时间。

算法分析

令 $r=\sum_{i=1}^{n} b_i$，$q=\sum_{i=1}^{p} g_i$。

对于第 i 个人，我们需要记录其当前开始处理时间 $last_i$ 和下次空闲时间 $free_i$。

对于时刻 t，我们把满足 $free_i \le t$ 的空闲的人拿出来，按 $last_i$ 为第一关键字编号为第二关键字从小到大进行排序以得到空闲的人的选择顺序。

第 i 个人选择时，按优先级从高到低访问其主题列表，找到一个已到达的请求，修改 $last_i$ 和 $free_i$，同时修改对应主题已处理的请求个数。

我们发现，只有某个请求到达或者某个人处理完某个请求开始空闲的时刻是可能进行选择的，而这样的有效时刻不超过 $2r$ 个。

为了找出这样的有效时刻，我们可以用一个优先队列来进行实时维护，队头即为下一个有效时刻。

当队列为空时处理完毕，最后一个有效时刻即为答案。

时间复杂度为 $O(r(p\log p+q+\log r)+q\log r)$，空间复杂度为 $O(r+q)$。

71 1999 A Bee Breeding

题目大意

蜂巢图案如题中图所示，巢室顺时针编号。求给定的一对巢室 $a,b(1 \le a,b \le 10000)$ 之间的最短距离。

算法分析

令 $c=10000$ 表示巢室的最大编号。

以巢室 1 为原点，水平方向为 x 轴建立平面直角坐标系，相邻巢室之间的坐标差为 $(\pm1,\pm1)$ 或 $(0,\pm2)$，则我们可以预处理出前 c 个巢室的坐标。

设巢室 a 坐标 $(x1,y1)$，巢室 b 坐标 $(x2,y2)$，则易知答案为 $|x1-x2|+\max\{\frac{|y1-y2|-|x1-x2|}{2},0\}$。

时间复杂度为 $O(c)$，空间复杂度为 $O(c)$。

72 1999 C A Dicey Problem

题目大意

给出一个 $n\times m(1\le n,m\le 10)$ 的地图和一个六面骰子的起点及其顶面和正面的数字，骰子可以向相邻的四个格子中写着其顶面数字或 -1 的格子滚动。求骰子从起点出发回到起点的最短路径，保证方案唯一。

算法分析

已知顶面和正面即可确定骰子形态，可以打表存储其余四个面，那么骰子滚动一格后新的形态也可以据此算出。

接下来用 BFS 求出骰子以某种形态到达某个位置的最短路径即可。

时间复杂度为 $O(mn)$，空间复杂度为 $O(mn)$。

73 1999 D The Fortified Forest

题目大意

给出 $n(1 \le n \le 15)$ 棵树的位置、价值及其砍倒后能做栅栏的长度，要求砍掉最小总价值的树使得其做成的栅栏足以把剩下的树围起来，并且要使砍掉的棵数最小。

算法分析

我们可以用 DFS 来枚举哪些树要砍，对剩下的树用 Graham 扫描法求凸包，若凸包周长不超过砍掉的树能做成的栅栏总长即合法，取总价值最小的前提下棵数最小的一组解即可。

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

74 1999 E Trade on Verweggistan

题目大意

有 $w(1 \le w \le 50)$ 堆物品，第 i 堆自上到下有 $b_i(0 \le b_i \le 20)$ 个成本分别为 $c_{i,j}$ 售价为 10 的物品，每堆物品只能自上到下依次买入。求最大利润及可能买入的物品个数。

算法分析

令 $s = \sum_{i=1}^{w} b_i$。

对于每堆物品，买到某个物品为止的利润是已知的，故每堆的最大利润是确定的，将其累加即得总最大利润。

而每堆可以得到最大利润的可能物品个数也是已知的，要求可能的总买入物品个数即转化为背包问题。

时间复杂度为 $O(s^2)$，空间复杂度为 $O(s)$。

75 1999 G The Letter Carrier's Rounds

题目大意

详见原题。

算法分析

按题意模拟即可，借助 C++ 的 map 进行判重。

注意地址间存在多余空格。

时间复杂度为 $O(n \log n)$，空间复杂度为 $O(n)$。

76 1999 H Flooded!

题目大意

给出一个 $m \times n(1 \le m, n \le 30)$ 的区域，每块有一个高度，高的块的积水会往低的块流。求给定总积水量对应的积水高度和积水块数。

算法分析

我们将各块高度从小到大排序，逐层判断当前水量能否淹没更高一层的块即可。

时间复杂度为 $O(mn \log mn)$，空间复杂度为 $O(mn)$。

77 1998 B Flight Planning

题目大意

飞机的实际速度为给定空速和关于飞行高度线性变化的风速的和，每小时消耗的燃料随着飞行高度与最佳高度差的绝对值的增加而增加，飞机的瞬间升高也需要升高高度对应的燃料，要求飞行高度均为整数且可行的飞行高度区间长度为 $l = 20$。给出连续 $n(1 \leq n \leq 100)$ 个航段的长度，求在每一航段的飞行高度使得消耗总燃料最少，消耗总燃料相同时要求飞行高度的字典序最小。

算法分析

容易想到用动态规划来解决这个问题。令 $f(i, j)$ 表示到第 i 段飞行高度为 j 消耗的最少总燃料，倒过来转移即可保证字典序最小。

时间复杂度为 $O(l^2 n)$，空间复杂度为 $O(ln)$。

78 1998 C Lead or Gold

题目大意

有 $n(1 \leq n \leq 100)$ 种由三种物质组成的混合物，第 i 种混合物的组成比例为 $a_i : b_i : c_i$。判断能否由这些混合物得到组成比例为 $a_0 : b_0 : c_0$ 的产物。

算法分析

对于第 i 种混合物的组成比例 $a_i : b_i : c_i$，我们可以将其表示为平面上的点 $(\frac{a_i}{a_i+b_i+c_i}, \frac{b_i}{a_i+b_i+c_i})$。

那么，由两种混合物 $(x1, y1)$ 和 $(x2, y2)$ 得到的产物显然为 $\lambda(x2, y2) + (1-\lambda)(x1, y1)$，即 $(x1, y1) + \lambda((x2, y2) - (x1, y1))(0 \leq \lambda \leq 1)$，故产物在这两点间线段上。

同理，当有三种混合物时，两种混合物线段上任意一点都与第三个点组成线段，合起来为一个三角形。以此类推，所有混合物得到的产物为这些点的凸包内任意一点。

故我们首先用 Graham 扫描法求出凸包，再判断点 $(\frac{a_0}{a_0+b_0+c_0}, \frac{b_0}{a_0+b_0+c_0})$ 是否在凸包内即可（当然若凸包小于 3 个点要特判）。

时间复杂度为 $O(n \log n)$，空间复杂度为 $O(n)$。

79 1998 D Page Selection by Keyword Matching

题目大意

每个网页和查询有不超过 $n = 8$ 个长度不超过 $l = 20$ 的英文关键字（不区分大小写），每个关键字的重量按出现次序从 n 开始依次减一，一个查询与一

个网页的关系强度为匹配关键字在两者的重量积之和。穿插给出 $p(1 \le p \le 25)$ 个网页和 $q(1 \le q \le 5000)$ 个查询，对于每个查询求出在其之前出现的与其关系强度最高的 5 个网页的编号。

算法分析

将网页的关键字全部转成小写后存储，出现查询时与之前的每个网页的每个关键字逐个进行匹配，将网页按关系强度排序后输出即可。

时间复杂度为 $O((ln^2p + p\log p)q)$，空间复杂度为 $O(lnp)$。

80 1998 E Petri Net Simulation

题目大意

给出一个 $n(1 \le n \le 100)$ 个有初始非负权值的点的网络和 $m(1 \le m \le 100)$ 种变迁，每种变迁发生一次使该种变迁的输入列表里每个点权值减 1，输出列表里每个点权值加 1（重复出现加减多次），若会导致出现权值为负的点则该变迁不允许发生。判断能否完成 $t(1 <= t <= 1000)$ 次变迁并求出结束后每个点的权值，保证答案唯一。

算法分析

按题意模拟即可。

时间复杂度为 $O(mnt)$，空间复杂度为 $O(mn)$。