1.聪明的小偷

(thief.pas/c/cpp)

【问题描述】

从前有一个收藏家收藏了许多相同的硬币，并且将它们放在了 n 个排成一排的口袋里，每个口袋里都装了一定数量的硬币。

这些硬币价值不菲，自然引起不少人觊(ji,4)觎(yu,2)，于是收藏家每天都会来检查一次这 n 个口袋，首先他会先检查每个口袋是不是都有硬币，之后他会计算出第 1 个和第 2 个口袋的硬币数量之和，第 2 个与第 3 个口袋的硬币数量和，如此直到第 n-1 个与第 n 个口袋的硬币数量之和，得到 n-1 个数的序列。

如果收藏家发现某个口袋没有硬币，或者他计算得到的序列较上一天相比有变动，那么收藏家就知道肯定有人动了他的硬币。

有一个聪明的小偷，他想在收藏家不知道的情况下偷走一些硬币，为此，他不仅可以偷偷地从某个口袋中拿出一些硬币，也可以将硬币在口袋间任意移动，现在他想知道对于给定的 n 个口袋及对应的硬币数量，他最多能拿多少枚硬币。

小偷是很聪明的，他早就算出来啦，但是他想考考作为徒弟的你……

【数据规模与约定】

对于 50%的数据，有 2≤n≤20，每个口袋中硬币数量≤20。

对于 100%的数据，有 2≤n≤999，每个口袋中硬币数量≤10000 且为正整数。

【问题分析】

通过观察可以很容易地发现，当 n 为偶数时，答案一定为 0，当 n 为奇数时，答案为所有编号为奇数的口袋中硬币数量的最小值-1。

证明也很简单，考虑最终 1 号口袋相对于初始时的硬币数量增减，则 2 号口袋硬币数量的增减必须抵消 1 号口袋的变动，类似地，3 号口袋必须抵消 2 号口袋的变动，之后考虑 n 个口袋中硬币总数量的变化，就可以得到该题的结论。

2.提前

(forward.pas/c/cpp)

【问题描述】

这个题目标题的意思显然不是说这次模拟赛会提前结束或者会在考试结束前提前告知 solution 等等——而是用来描述在序列上的一个操作。

现在有一个长度为 n 的数列，第 i 个数字即为正整数 i，在这个序列上只有一种操作：提前操作——将一个数从其当前的位置提到序列的最前面，在该数之前的数依次后移一位，比如在序列 1 2 3 4 5 上将数字 4 提前，那么就会得到序列 4 1 2 3 5，之后如果再将数字 2 提前，那么便得到序列 2 4 1 3 5。

现在给定一系列的提前操作，你的任务就是执行这些操作并且输出执行这些操作之后最终的序列。

【数据规模与约定】

对于 50%的数据，有 1≤n,m≤2000。

对于 100%的数据，有 1≤n,m≤50000，1≤T≤6。

【问题分析】

我们考虑从最终序列的首位开始填补数字：

逆序扫描操作序列，如果遇到一个之前没被扫描过的操作数，那么就把它填入当前未填序列的首位。

扫描完成后，将剩余未填入的数按从小到大的顺序依次填入未填序列即可。

3.采油

(oil.pas/c/cpp)

【问题描述】

一家石油公司最近在一片区域开采石油,这片区域可以看作一个二维平面，其中 x 轴平行于地面，x 坐标表示地表的水平位置，而 y 轴垂直于地面，y 坐标表示深度。这片区域内还有 n 个石油矿，每个石油矿都可以看作一条平行于 x 轴的线段，其中石油的储量即为该线段的长度，如下图就对应着样例一的输入：

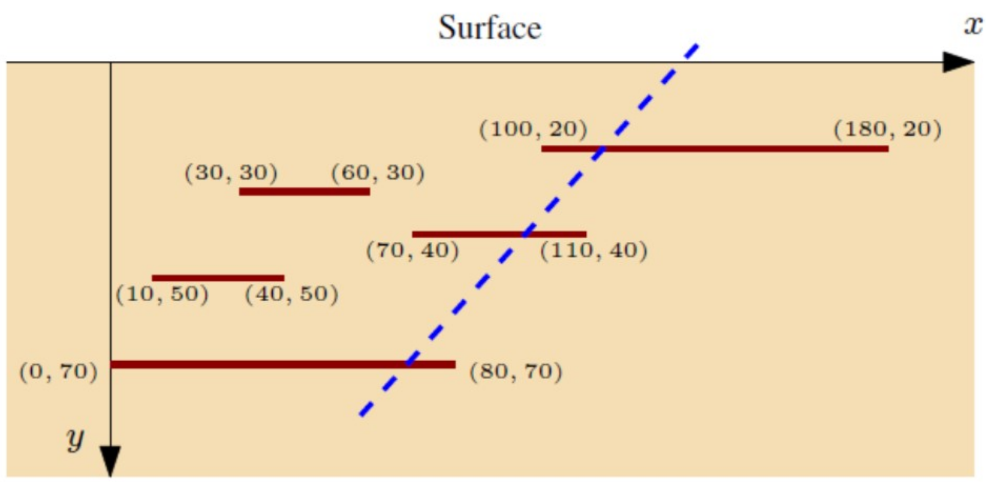


Figure G.1: Oil layers buried in the earth. This figure corresponds to Sample Input 1.

现在这家石油公司想要挖一个石油井,石油井可以看作一条与 x 轴相交的直线（如上图中蓝色虚线所示），通过挖石油井，这家石油公司能获取到所有与石油井有公共点的石油矿内的石油（端点处或者在线段内均算）。

你的任务就是替这家公司找到一个挖石油井的方案,使得其能获得最大储量的石油，为了方便，你只需要输出最大储量即可。

【数据规模与约定】

对于 20%的数据，有 $n \leqslant 2$。

对于 50%的数据，有 $n \leqslant 100$。

对于 100%的数据，有 $1 \leqslant n \leqslant 1000$，$|x1|,|x2| \leqslant 1000000$，$1 \leqslant y \leqslant 1000000$。

【问题分析】

很显然，一定存在一种最优解，使得其经过至少两个端点，因此可以直接在 2n 个端点中枚举直线，并判断每条直线穿过的线段的长度之和，这样做时间复杂度是 $O(n^3)$的，即可以得到 50 分。

为了优化，我们可以枚举基准点。对于枚举的每一个基准点，考虑过这个基准点的直线的斜率倒数 1/k，在基准点看来，每条线段都对应着一个 1/k 的闭区间，表示当 1/k 的值落在该范围内的时候，直线会与这条线段有公共点。

这样就把问题转化为选取一个点使得包含它的线段的权值之和尽可能大,这是一个非常经典的问题，可以使用多种方式完成。

算法总时间复杂度 $O(n^2 \log n)$。

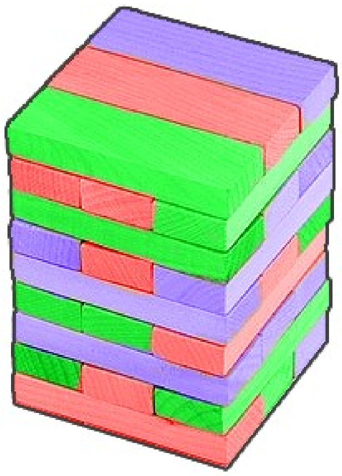
4.叠叠乐

（jenga.pas/c/cpp）

【问题描述】

有一个叫 jenga 的游戏，中文名叫作叠叠乐，如下图。

（叠叠乐的木板有红、绿、蓝三种颜色，放置的时候是一层横一层竖下来的。）



这个游戏只有 1 个人玩！这个人有一个 6 面骰子，掷出每个面的概率相等，六个面分别是绿、绿、蓝、蓝、红、黑。

现在这个人掷骰子，如果朝上的那一面不为黑，那么其在该回合可选择一块相应颜色的木板抽出并放到顶上，否则其停一回合。

抽木板的规则：

1、不能从当前的最顶层抽木板，无论其是否已经完成。

2、抽完木板之后这个结构必须是平衡的，即每层 123 三个位置，或者是 2 位置有一块木板，或者是 1 和 3 位置都有木板。

3、放置木板的时候，如果顶层三个位置未填满，那么必须先填满顶层才能开始下一层。

4、如果在以上三条规则下该回合没有木板可抽，那么停一回合，否则必须抽一块相应颜色的木板。

如果三种颜色的木板都不能抽了，那么游戏结束。

现在给定一个叠叠乐的初始局面，求在操作者争取总用时最少的情况下，游戏结束时，掷骰子的期望次数。

【数据规模与约定】

对于 15%的数据，$n = 2$，

对于 30%的数据，$n \le 3$，

对于 50%的数据，$n \le 4$，

对于 70%的数据，$n \le 5$，

对于 100%的数据，$2 \le n \le 6$，给定的叠叠乐游戏初始局面合法，真实答案与要求输出的 4 位小数之差的绝对值小于 0.000049（比如样例中的答案事实上为 17.119213696601999，与要求输出的 4 位小数 17.1192 之差为 0.000013696601999，小于 0.000049。

【问题分析】

这题本身模型不难，直接进行记忆化搜索算出每个状态的答案即可。

这题的关键在于状态数量的优化，有几个比较关键的优化点（其余一些细节可以参考标程）：

1、顶层以下所有的层都是无序的。

2、每层只需要记录能抽出的木板的状态即可，如果一层不能抽出木板了，那么就可以在状态表示中删掉这一层。

3、每层中的两侧是对称的。

实现时，将状态进行 hash 判重，或者使用 STL 实现应该也可以（没试过）

做完以上优化后，对于一个 6 层的叠叠乐游戏，在标程中不同的状态数量不超过 20W，足以在出题人的机子上 1s 内求得结果。