1 购票方案 (buy)

考虑如何求出 $sum_1, ..., sum_N$，一个很显然的性质是 sum_i 单调不减。

若当前考虑到第 i 次进入游乐园，我们尝试枚举这次进入游乐园用的是哪一种类型的票，若枚举的使用的票的时长为 num_j，我们求出一个最早的 k 使得从第 k 次进入游乐园开始使用这张票能用到第 i 次进入游乐园。若我们能在均摊 $O(1)$ 的时间复杂度内求得 k，就能在 $O(NK)$ 的复杂度内完成这道题。

现在我们考虑如何求出 k，考虑对于每种不同的票 j，随着枚举的 i 的增加，k 也是单调增加的，所以我们就不用从头开始枚举 k 而是从上一个 k 的位置开始枚举从而实现均摊 $O(1)$ 的复杂度。

时间复杂度 $O(NK)$。

2 直线相交 (intersect)

首先考虑一些基本的事实：若当前有 N 根木棒，则可能构成的最多的交点数为 $N\times(N-1)/2$，最少构成的交点数为 0（N 根木棒彼此平行）。

考虑一个正着的 dp：记 $f_{i,j}$ 表示当前考虑了前 i 根木棒（前 i 根木棒和后面的都不平行），是否有可能构成 j 个交点。转移时枚举下一组相互平行的木棒数量 k，则有：

$$f_{i,j}=\cup_{k=1}^{i}f_{i-k,j-(i-k)\times k}$$

初始状态为 $f_{0,0}=$True。f 的第一维大小为 N，第二维大小为 $O(N^2)$，转移的复杂度是 $O(N)$ 的，总复杂度 $O(N^4)$，可以通过 80% 的数据。

接下来考虑倒过来处理这个问题：研究当前的方案和全部两两相交的方案相差了多少。如果某方案的木棒可以分成 k 组，每组分别是 $i_1,i_2,...,i_k$ 个，那么这个分组方案和全满的分组方案相差了 $\sum_{j=1}^{k}i_j\times(i_j-1)/2$ 个交点。据此，考虑另一种 dp 方案：

记 g_i 表示与全满状态相差 i 根木棒至少需要多少根木棒，我们有：

$$g_i=\min_{j=1}^{N}j+g_{i-j\times(j-1)/2}$$

这样总 dp 的复杂度就是 $O(N^3)$ 的了。考虑一组询问 (N,M)，他能完成当且仅当 $g_{N\times(N-1)/2-M}\le N$。总复杂度为 $O(N^3+M)$。

3 记录数列 (record)

我们首先考虑在什么情况下可能存在阶梯数列的原数列：若阶梯数列的第一位是 k，那么阶梯数列的前 k 位必然也要是 k，我们把这 k 位记为一段。顺着这个思路继续考虑第 $k+1$ 位... 一直考虑完整个数列。假设这个数列一共被分成了 l 段，我们记 $num_1, ..., num_l$ 为每段的长度。

接下来考虑一个容斥 dp：记 $f_{i\ j,0/1}$ 表示考虑完第 i 段，存在至少 j 个相邻的段是可以接上的（即可以被看做一个更大的段），第 i 段序列是单调增的 0 还是单调减的 1，此时可能的序列数（忽略组合系数）。有转移方程：

$$f_{i,j,0} = f_{i-1,j,0} + f_{i-1,j,1} + f_{i-1,j-1,0};$$

$$f_{i,j,1} = f_{i-1,j,0} + f_{i-1,j,1} + f_{i-1,j-1,1}.$$

在考虑组合系数后，N 段至少 j 个相邻的段的表达式为：

$$sum_j = (f_{N,j,0} + f_{N,j,1}) \times (N-j)!$$

最后的答案为：

$$Ans = \sum_{i=0}^{N} sum_i \times (-1)^i$$

如果不考虑 $a_i = 1$ 的情况，这个 dp 是对的（因为当 $a_i = 1$ 时它可能即属于被用于单调增的连续段和单调减的连续段）。当面对 $a_i = 1$ 时，我们需要在 dp 中加一维额外的标记表示当前是否在处理可以接上升段还是下降段的（具体实现见代码）。

时间复杂度 $O(N^2)$。

4 随机游走 (walk)

首先考虑一个子问题：如何求出每个节点期望经过了多少次？

考虑高斯消元，我们记 deg_i 为 i 号节点的度数，p_i 为初始时位于 i 号节点的概率，f_i 为 i 号节点的总经过次数（包括初始），则我们有：

$$f_i = \sum_j f_j/deg_j + p_i, \quad (1 \le i \le N-1)$$

其中 j 是所有和 i 相邻的节点，$f_n = 0$。现在有个 N 个未知数和 N 个方程，可以通过高斯消元在 $O(N^3)$ 的时间内完成一组初始概率的求解。

如果我们记高斯消元的系数矩阵为 A，则方程可以写成：$Af = p$。由于对于不同的初始概率 A 矩阵是相同的，求解过程一致，因此我们可以采用一个 trick：记 p^i 为第 i 次对初始权重修改后得到的概率，可以把所有的 p 拼在一起得到矩阵 $B = [p^0, p^1, ..., p^N]$（这种修改不会超过 N 个），再解方程 $Af = B$。这样一次能解出所有初始权重的结果且复杂度为 $O(N^3)$。

我们更进一步地来考虑如何求出每条边期望经过了多少次？

由于期望的线性性，我们可以得出对于一条边 (u, v)，经过这条边的期望次数为 $f_u/deg_u + f_v/deg_v$，即期望在这条边上获得的分数为 $(f_u/deg_u + f_v/deg_v) \times w_{(u,v)}$。同时可以发现对于边权的修改，可以在 $O(1)$ 时间内完成。

总时间复杂度为 $O(N^3 + NM + Q)$