T1

记 $sz[u]$ 表示 u 的子树大小，$cnt[u]$ 表示初始掉落在 u 上的球数，$S[u]$ 表示初始掉落在 u 子树内的球数，$mx[u]$ 表示初始掉落在 u 祖先上的球数。显然，如果存在一个 $sz[u] < S[u]$，答案为 0。

考虑进行一个树形 DP。设 $dp[u][i]$ 表示从 u 的祖先（不包括 u）掉落下 i 个球，最终在 u 子树内的方案数。这 i 个球是已经选定了编号的，在之后我们只需要假定他们来自外面，且互不相同，而不需要关心具体是什么。显然 $0 \le i \le \min(mx[u], sz[u] - S[u])$。

当 u 是叶子时，所有 $dp[u][i] = 1$。

当 u 只有一个儿子时，讨论子树内是否堆满（最终是否有球留在 u 上），也即是否有 $i + S[u] = sz[u]$。如果堆满，可以事先从 $cnt[u] + i$ 个球中选择一个留下，其余往下传；否则全部往下传。

当 u 有两个儿子时，我们不妨假设 u 是 u 父亲的右儿子，x 是 u 的右儿子，y 是 u 的左儿子。如果堆满，需要选出一个球留下，再选出 $sz[x]$ 个球放到 x 中；否则，分为两种情况，① x 中堆满，y 中既有 $cnt[u]$ 中的球，也有 i 中的球，② x 中未堆满，y 中只有 $cnt[u]$ 中的球。那么只要枚举一下具体是 $cnt[u]$ 中的多少个球进了 y（或者 i 中的多少个球进了 x）就行了，转移系数是组合数。

$u = 1$ 时可能需要一些特殊处理，不过都是平凡的。

T2

首先注意到形成的图去掉方向后一定是森林，因为如果存在环 $u_1 \xrightarrow{e_1} u_2 \xrightarrow{e_2} u_3 \cdots u_k \xrightarrow{e_k} u_1$，假设边 e 对应的边权为 $w(e)$，那么 $w(e_k) > w(e_{k-1}) > \cdots w(e_1) > w(e_k)$，矛盾！

且这个森林中每个弱联通块恰好存在一对点 (u, v) 使得存在边 $u \to v, v \to u$，那么连通块的个数就恰好等于这样的点对数。

因此考虑上二项式反演，钦定 t 个点对。此时我们把这 t 个点对对应的 t 条无向边按大小从小到大排个序，假设第 i 小的边为 e_i，然后把图中 $\frac{n(n-1)}{2}$ 条边分成一下三类：

- 两个端点为同一个点对，即 t 条无向边中的一个。
- 两个端点属于不同点对，假设对应的边分别为 e_i, e_j，那么这条边的边权 w_0 满足 $w_0 < w\left(e_{\min(i,j)}\right)$
- 两个端点中有一个属于某一个点对，假设对应的边为 e_i，那么这条边的边权 w_0 满足 $w_0 < e_i$
- 两个端点都不属于任何点对，那么这条边的边权任意。

那么可以发现，对于排名为 i 的边 e_i，有 $n - 2t + (t - i) \times 4$ 条边的限制为边权 $< w(e_i)$

而图合法的充要条件即为以上限制全部成立。总的有条件约束的边数为 $\sum_{i=1}^{t} (n - 2t + (t - i) \times 4 + 1) = 2tn - 2t^2 - t$

因此当前条件下符合条件的图个数为

$$
\begin{aligned}
&\binom{\frac{n(n-1)}{2}}{2tn - 2t^2 - t}\left(\frac{n(n-1)}{2} - \left(2tn - 2t^2 - t\right)\right)! \prod_{i=1}^{t} \binom{\sum_{j=1}^{i} (n - 2t + (t - j) \times 4 + 1)}{n - 2t + (t - i) \times 4} \\
=&\left(\frac{n(n-1)}{2}\right)! \cdot \frac{1}{(2tn - 2t^2 - t)!} \cdot \frac{\left(2tn - 2t^2 - t\right)}{\prod_{i=1}^{t} i (2n - 2i - 1)} \\
=&\left(\frac{n(n-1)}{2}\right)! \cdot \frac{(2n - 2t - 3)!!}{t!\,(2n - 3)!!}
\end{aligned}
$$

而选 t 个点对并排序的方案数为 $\binom{n}{2t} \cdot (2t - 1)!! \cdot t!$

因此答案为

$$
\begin{aligned}
&\frac{1}{\left(\frac{n(n-1)}{2}\right)!} \sum_{t=k}^{\lfloor \frac{n}{2} \rfloor} (-1)^{t-k} \binom{t}{k} \binom{n}{2t} (2t - 1)!! \cdot t! \cdot \left(\frac{n(n-1)}{2}\right)! \cdot \frac{(2n - 2t - 3)!!}{t!\,(2n - 3)!!} \\
=&\sum_{t=k}^{\lfloor \frac{n}{2} \rfloor} (-1)^{t-k} \binom{t}{k} \binom{n}{2t} (2t - 1)!! \cdot \frac{(2n - 2t - 3)!!}{(2n - 3)!!}
\end{aligned}
$$

复杂度 $O(n)$。

T3

3.1 测试点 1,2,3

测试点 1 的范围较小，直接暴力枚举删掉哪些边即可。

测试点 2 3 范围稍大，直接暴力枚举不可行。

优化搜索，删掉最短路上的边。

测试点 2 大约需要运行 10 秒左右。

测试点 3 大约需要运行 10 分钟左右。

3.2 测试点 4

边数虽然很多，但是删边个数没有限制。

即我们可以保留最后的那条最短路，将其余的边都删除。

所以也就是求这个图的从点 1 到点 N 的最长路。

参考解法：点数较少，只有 20 个点，因此可以用状态压缩动态规划解决。

该测试点为测试点 4 的扩展。

3.3 测试点 5

将节点分为 50 块，每一块中都是 20 个节点。

相邻块之间用一条边相连。

参考解法同测试点 4，对每一块用状态压缩动态规划求最长路。

3.4 测试点 6，7

这两个测试点类似测试点 5，每个块只有 10 个点和 20 条边，但是删边个数有限制。

参考解法：

对每个块搜索，处理出删掉 k (0 k 20) 条边后，从块内起点到块内终点的最短路的最大值。

然后再进行动态规划，F[i][j] 为前 i 块中删掉了 j 条边，从节点 1 到第 i 块的终点的最短路的最大值。

3.5 测试点 8

该测试点是一个 100 行 100 列的网格图，边权都是 1，删边个数无限制。

起点左上角，终点右下角。

可以直接构造出一条长度为 9998 的路径。

证明：不存在长度为 9999 的路径。

3.6 测试点 9

一个 2 行 1000 列的网格图，且已经删掉了若干条边。

起点左上角，终点右下角。

动态规划求最长路。

3.7 测试点 10

10000 个点的图，边权都是 1 。

图中存在从点 1 到点 10000 的哈密尔顿路。