1 克莱因瓶 (klein)

可以固定住一个点, 到另一个点如果经过了边界, 则可以认为是将点做类似"镜面翻折" 的操作. 这样只需要检查 $O(1)$ 个翻折后的情况就能算出答案了.

2 马克思么克斯 (maxmex)

显然 $f(k)$ 是单调不减的, 我们对每个 v 考虑如何找到最大的 k 使得 $f(k) < v$.

对 v 从小到大扫描, 考虑 v 所在的所有位置将序列分割成若干区间, 如果一个区间不包含 v, 也即它落在一个区间里.

那么 $f(k) < v$ 就等价于说, 把所有 $< v$ 的值所构成的这些区间拿出来, 对于任何长度为 k 的区间, 都在某个区间之中.

在几何上看, 我们加入一个区间就是加一个矩形, 我们希望知道目前矩形的并里可以放的最高的一条斜率为 1 的直线的高度, 那么我们只需要维护这些矩形的并构成的 "边角", 用一个堆来存, 就可以做到总时间 $O(n\log n)$ 了.

3 周游世界 (tour)

考虑枚举走的第一条边. 假设走了第一条边长度是 e, 那么问题转化成接下来走的边乘积不超过 $\lfloor m/e\rfloor$. 我们可以先把接下来不超过 m/e 的边都抠出来, 解决剩下的这个 $\lfloor m/e\rfloor$ 级别的子问题. 如果我们能在 $T(m)$ 的时间内解决单个起点的问题, 就能在 $O\left(\sum_{i=2}^{m} T(m/i)\right)$ 的时间内解决完整的问题.

注意到一条边长度为 e 的话只会进行 m/e 次更新, 这样就有 $T(m) = O(m\log m)$, 总复杂度 $O(m\log^2 m)$.

考虑乘积下取整的转化, 我们到达一个点的路径的权值乘积 p 其实并不需要记录, 而是记录 $\lfloor m/p\rfloor$ 的值. 这个下取整只有 $O(\sqrt{m})$ 种可能性. 对于 e 这条边进行的更新, 所有 $\lfloor m/ex\rfloor$ 只有 $\sqrt{m/e}$ 种取值, 这样我们通过一点技巧将更新的复杂度压到 $O(\sqrt{m/e})$ 之后, 总共复杂度就变成了

$$T(m) = O\left(\sum_{e=2}^{m}\sqrt{m/e}\right) = O(m).$$

总共复杂度就是 $O(m\log m)$ 了.

4 大度一点 (big)

有一些容斥做法可以求单点做到 $O(2.414^n)$ 甚至 $O(2.618^n)$, 但和下述方法关系不大, 故略.

考虑通过下述过程发掘图的结构: 如果一个图存在一个度数 ≤ 1 的点, 我们就将其删去. 容易发现, 无论按照什么样的顺序删去, 我们最后得到的不需要删点的图都是同一个图 C. 直观上看, 任何图 G 都被上述过程得到了一个分解: 任何一个图 G 被唯一表示成 C 上的每个点 "生长出" 一颗树, 再加上若干颗树作为额外的连通块. 接下来我们记删去那些额外连通块之后得到的图为 $NoForest$.

我们记 g_S 表示点集的子集 S 所对应的无向图的数量, 显然有 $g_S = 2^{|E[S]|}$, 其中 $E[S]$ 表示 S 的导出子图的边集. 那么我们有

$$g_S = \sum_{A \sqcup B = S} forest_A \cdot noForest_B,$$

其中 $A \sqcup B$ 表示无交并. 容易在 $O(3^n)$ 的时间内计算出 $tree_S$ 进而在 $O(3^n)$ 的时间推出$forest_S$, 其中 $tree_S$ 是生成树的数量. 所以根据上式, $noForest_S$ 也是可以在 $O(3^n)$ 时间内求得的.

接下来我们定义 $g_{i,S}$ 表示编号 $\ge i$ 的点都在 C 中的图的方案数, 那么当 $i+1$ 变到 i 的时候, 对于 $i \notin S$ 我们显然没有事情要做, 否则需要考虑 i 号点不在 C 中的情况, 此时 i 恰有一个子树, 我们有

$$g_{i,S} = g_{i+1,S} - \sum_{\substack{A \sqcup B \\ i = \max A}} tree_A \cdot g_{i+1,B} \cdot |E[i,B]|,$$

其中 $E[i,B]$ 表示 i 到 B 之间连的边. 容易发现, 第 i 轮计算的总复杂度是 $O(3^i 2^{n-i})$, 所以求和之后复杂度还是 $O(3^n)$, 整个问题在 $O(3^n)$ 时间内解决.

使用子集卷积优化上述的每一步, 可以做到 $O(2^n n^3)$. 也有一些其他的容斥做法做到 $O(3^n)$, 但可能不太容易优化.