



集你太美 解题报告

长沙市雅礼中学 邓宇帆

题目大意

给定一张 n 个结点的无向完全图 G , 边 (i, j) 带非负整数权 $v_{i,j}$, 保证 $v_{i,i} = 0, v_{i,j} = v_{j,i}$.

同时, 每个结点有一个非负整数变量 w_i .

定义对一个点 i 进行一次 **收集操作** 为, 将 w_i 的值加上 $\sum_j v_{i,j}$, 并将 w_j 的值减去 $v_{i,j}$. 称一次对点 i 的收集操作 **合法**, 当且仅当操作前 $w_j \geq v_{i,j}$.

在图 G 上称一组点权 **收集-free**, 当且仅当以这组点权为初始状态, 存在一种方式, 能够进行 **无限次** 合法的收集操作.

你有 **两种** 任务. 第一种, **构造** 一组点权 w'_i , 使得 w'_i 收集-free, 且 **最小化** $\sum_i w'_i$; 第二种, 给定一组点权 w'_i , 你需要 **判断** w'_i 是否 收集-free.

数据范围

令 o 表示任务类型, m 表示边权非 0 的边数.

对于所有数据, $o \in \{1, 2\}, 1 \leq n \leq 3 \times 10^5, 0 \leq m \leq \min(10^6, \frac{1}{2}n(n-1)), 0 \leq w'_i \leq 10^{18}$.

Subtask 编号	n 的上界	m 的上界	特殊性质	分值
1	10	10	/	10
2	20	100	/	10
3	300	300	/	10
4	/	$n - 1$	$o = 1$, 非 0 边构成一棵树	10
5	/	/	$o = 1$	30

Subtask 编号	n 的上界	m 的上界	特殊性质	分值
6	/	/	$o = 2$	20
7	/	/	/	10

解题过程

部分分

$$n \leq 10$$

不难发现对于任意一组 收集-free 的点权, 一定存在一种方式使得在操作了每个结点各一次后点权与初始情况相同. 故枚举每个排列后倒推点权或判断是否合法即可. 时间复杂度 $O(n! \text{poly}(n))$.

$$n \leq 20$$

这部分留给一些 $O(2^n \text{poly}(n))$ 的 Dp 做法.

$$n \leq 300$$

这部分留给具有奇思妙想的选手.

$o = 1$, 非 0 边构成一棵树

随意定一个根, 每个结点的点权设为非返祖边边权之和, 按深度从大到小操作即可.

约定

用 **局面** 指代 w_i 序列. 称局面 A 是局面 B 的 **子局面**, 当且仅当局面 A 中每个点的点权都不超过局面 B 中对应点的点权.

我们不再认为 $v_{i,j}$ 是一条边的边权, 而是连接了点 i 与点 j 的边权为 1 的边的 **数量**. 显然这不改变收集操作对局面的影响. 故下文假定每条边的边权都为 1, 并且不再强调边权.

若 G 不连通, 则显然构造点权可以对每个连通块分别构造, 并将不在这个连通块的点权设为 0;

判断一个局面是否 收集-free 可以对每个连通块分别判断. 故下文假定 G 连通.

结论

一个局面 A 收集-free 的 **充要条件** 为, 图 G 存在一种 DAG 定向的方式, 使得将定向后每个结点的入度看作一个局面 B 后, 有 B 是 A 的子局面. 下文中给出了两种证明方式, 其中第一种较简单, 而第二种提供了一个优美的结构, 且能够推出更多的性质.

第一种证明

充分性: 我们每次可以对拓扑序最靠前的结点进行收集操作. 由于这个结点的所有邻边都是出边, 这次操作合法. 且操作完后所有邻边变为入边, 则这个结点在拓扑序中移动到最靠后的位置, 新图依旧是 DAG, 故可以进行无限次合法的收集操作.

必要性: 对于局面 A , 任取一个无限长的收集操作序列, 根据第二种证明中的 Lemma 1, 所有结点都在序列中出现. 根据在序列中 **第一次** 出现的位置将所有结点升序排列 (不妨假设排列后的结点顺序即为 $1, 2, \dots, n$), 我们声称这个拓扑序给出的局面 B 是 A 的子局面.

显然在结点 i 被操作前, w_i 不会增加, 且结点 $1, 2, \dots, i-1$ 都被至少操作过一次, 故这些收集操作合法的必要条件有 $w_i \geq \sum_{j=1}^{i-1} v_{j,i}$. 故 B 的确是 A 的子局面, 命题得证.

第二种证明

准备工作

我们先对收集操作进行观察. 发现收集操作合法的条件与所有非自身的点的点权有关, 较为复杂. 于是我们可以想到与收集操作对称的另一个操作, 称为扩散操作.

形式化地定义, 对一个点 i 进行一次 **扩散操作** 为, 将 w_i 的值减去 $\sum_j v_{i,j}$, 并将 w_j 的值加上 $v_{i,j}$. 称一次对点 i 的扩散操作 **合法**, 当且仅当操作前 $w_i \geq \sum_j v_{i,j}$.

容易发现扩散操作合法的条件只与自身的点权有关, 且扩散操作和收集操作互为 **逆操作**. 类似地, 用 **扩散-free** 指代一个局面存在一种方式能够进行无限次合法的扩散操作.

由于进行操作伴随着局面的变化, 故如果我们想考虑不同的局面之间的关系的话, 需要一个更好的结构来帮助思考.

容易想到可以对图 G 定义一张 **局面图** \mathcal{G} . 也即, 将每个不同的局面视为一个结点, 如果一个局面 A 能够通过 **一次合法的扩散操作** 变为另一个局面 B , 则 \mathcal{G} 中有 A 向 B 的 **有向边**.

构造总和最小的 收集-free 点权

我们先考虑 $o = 1$ 的情况.

类似准备工作中的思路, 我们希望把求解 收集-free 局面中 $\sum_i w_i$ 的最小值, 变为求解 扩散-free 的局面中 $\sum_i w_i$ 的最小值. 为此, 考虑在 \mathcal{G} 中找出哪些局面 收集-free, 以及哪些局面 扩散-free.

先考虑 收集-free 的局面, 对应着在 \mathcal{G} 上逆向行走. 则 收集-free 的局面有且仅有 强连通分量内的局面以及强连通分量能够到达的局面. 类似地, 扩散-free 的局面有且仅有 强连通分量内的局面以及能够到达强连通分量的局面.

Lemma 1. 如果一个局面 扩散-free, 那么在无限次进行合法的扩散操作的过程中, **所有点** 都会被操作无限次.

Proof. 假设存在一些点只会被操作有限次, 任取一个这样的点组成的连通块 G' . 显然 G 中至少有一个点会被操作无限次, 且 G 连通, 故 G' 一定有一个会被操作无限次的 **邻点** v .

记 $S = \sum_{i \notin G'} w_i$. 容易发现, S 只有在对 G' 内的点操作时才会增加, 故 S 只会增加有限次, 但每次对 v 进行操作时 S 都会减少, 故 S 会减少无限次. 又有 S 始终非负, 故矛盾.

Lemma 2. \mathcal{G} 中强连通分量内的局面, **出边** 指向的 **所有局面** 都在强连通分量内.

Proof. 取一个强连通分量内的局面 A , 以及它通过对点 u 进行扩散操作到达的另一个局面 B . 我们只需要证明 B 能够通过若干次扩散操作到达 A , 即可证明上述命题.

取一个经过 A 的简单环, 假设这个环的长度为 k , 从 A 开始沿着这个环走一圈所进行的扩散操作序列为 v_1, v_2, \dots, v_k . 根据 Lemma 1, $\{v_i\}$ 一定包含了 G 中的所有结点, 故存在一个下标 $1 \leq p \leq k$ 使得 $v_p = u$ 且对于任意的 $1 \leq i < p$ 有 $v_i \neq u$.

考虑一个新的长度为 k 的从 A 开始的操作序列 v' , 满足 $v'_1 = u$, 对于 $1 \leq i < p$ 有 $v'_{i+1} = v_i$, 对于 $p < i \leq k$ 有 $v'_i = v_i$. 现证明从 A 开始的操作序列 v' 合法.

显然只需证明前 p 次操作均合法即可. 由假设得第一次操作合法, 且剩下的 $p - 1$ 次操作都不是对于 u 进行的操作. 由于扩散操作合法只与进行操作的那个点的点权有关, 且点权越大越容易合法, 且如果不进行第一次操作, 依次进行剩下的 $p - 1$ 个操作合法, 故前 p 次操作合法.

显然从 A 开始按序列 v' 进行操作会回到 A 局面, 且从 A 开始操作 $v'_1 = u$ 会到达局面 B , 故局面 B 能通过若干次扩散操作到达 A , 命题得证.

根据 Lemma 2, 收集-free 的局面有且仅有 强连通分量内的局面. 进一步, 收集-free 的局面 **都是** 扩散-free 的局面.

这是符合我们在准备工作中的认识的, 因为扩散操作合法的条件显得更加宽松. 由此, 收集-free 的局面中 $\sum_i w_i$ 的最小值 **大于等于** 扩散-free 的局面中 $\sum_i w_i$ 的最小值.

于是, 我们可以开始考虑 扩散-free 的局面中 $\sum_i w_i$ 的下界.

由于局面会不断地改变, 我们希望将 $\sum_i w_i$ 对应到其他不会改变的量上面去.

为了方便对应, 我们认为每 1 单位的点权都是 **互不相同** 的. 对一个点 i 进行扩散操作时, 我们可以选取 i 已有的点权中的任意一部分, 并任意地将它们分配给其他点, 只要总的数量满足操作规则即可.

我们称一条边一定处于两种不同的状态之一, 分别为 **未分配** 和 **已分配**. 所有的边初始状态为未分配. 当我们对一个点 i 进行扩散操作时, 若 i 连接的边中存在一条边是未分配的, 则认为这次操作通过这条边移动到邻点的那 1 单位的点权被 **分配** 到了这条边上, 并将这条边的状态改为已分配; 若 i 连接的边中存在一条边是已分配的, 若这条边被分配的点权在 i 上, 则我们将这条边被分配的那 1 单位的点权通过这条边移动到邻点去; 否则任选 1 单位尚未被分配的点权移动过去.

显然操作了点 i 之后, 它连接的所有的边的状态都是已分配.

显然当 1 单位的点权被分配到了某一条边上之后, 这 1 单位的点权就只会在这条边的两个端点间通过这条边移动. 故每条已分配的边所被分配的点权 **互不相同**.

根据 Lemma 1, 在无限次进行合法的扩散操作的过程中, 所有点都会被操作至少一次. 故一定存在一个时刻, 在这个时刻之后每条边的状态都是已分配. 由于每条边被分配的点权互不相同, 则 扩散-free 的局面 $\sum_i w_i$ 的下界为 m .

由于收集-free 的局面中 $\sum_i w_i$ 的最小值大于等于扩散-free 的局面中 $\sum_i w_i$ 的最小值, 故收集-free 的局面 $\sum_i w_i$ 的下界也为 m .

现在考虑收集-free 的局面 $\sum_i w_i$ 能否取到 m . 容易发现, 如果我们对 G 中的每条无向边 **定向**, 使得定向后的图成为一个 **DAG**, 将每个点的点权设为它的入度, 则这个局面 收集-free.

具体地, 我们每次可以对拓扑序最靠前的结点进行收集操作. 由于这个结点的所有邻边都是出边, 这次操作合法. 且操作完后所有邻边变为入边, 则这个结点在拓扑序中移动到最靠后的位置, 新图依旧是 DAG, 故可以进行无限次合法的收集操作.

因此, 收集-free 的局面中 $\sum_i w_i$ 的最小值为 **边数**, 也即 $\sum_{i,j} v_{i,j}$. 具体的构造方法在上文中已给出. 显然能够做到线性的时空复杂度.

结论的证明

Lemma 3. 一个局面 A 收集-free 的充要条件为, 图 G 存在一种 DAG 定向的方式, 使得将定向后每个结点的入度看作一个局面 B 后, 有 B 是 A 的子局面.

Proof. 充分性在上文已提到, 下证必要性. 根据 Lemma 2, 收集-free 的局面都在 \mathcal{G} 的强连通分量内, 故收集-free 的局面都存在一种方式使得进行若干次收集操作后 **回到原局面**. 根据 Lemma 1, 在回到原局面前进行的这些操作中, 每个结点都会被操作至少一次. 故依旧可以为边分配 1 单位的点权, 且回到原局面后每条边都已分配.

只考虑 **被分配过** 的点权作为子局面, 我们尝试说明这样的子局面可以通过 DAG 定向的方式得到. 我们按照每个结点 **最后一次** 被收集操作的时刻来对每个结点排序, 并声称这个顺序也是 DAG 定向所参照的拓扑序.

显然对一个结点进行收集操作之后它的所有邻边被分配的点权都会来到这个结点上, 故此时在 DAG 定向的拓扑序中它可以位于最靠后的位置. 则通过倒推不难发现每个结点都在一个合理的拓扑序的位置上. 故这样的子局面可以通过 DAG 定向的方式得到.

判断点权是否 收集-free

根据结论, 我们只需要判断一个局面是否有一个子局面可以通过 DAG 定向得到. 下面给出一个简单的贪心算法.

显然这个子局面中一定有至少一个点权等于度数的结点, 对应 DAG 中没有出度的结点. 故可以尝

试在 G 中任取一个 **点权大于等于度数** 的结点, 将它所有的邻边的方向设为朝向这个结点, 并将这个结点从 G 中删去. 重复进行这一操作, 直到 G 为空 或者 G 非空但无法找到任何一个点权大于等于度数的结点.

假设我们任取的这个点权大于等于度数的结点在 DAG 定向中并非没有出度的结点, 则选取的这个结点一定有一些邻边被定向成朝向对侧的结点. 则不妨将这些邻边的方向改为朝向自己, 显然对侧的结点的入度变小, 且选取的这个结点始终合法, 故新的定向方案同样合法. 故将我们选取的这个结点钦定为 DAG 定向中没有出度的结点 **不劣**.

因此, 当无法继续选取结点时, 若 G 为空则原点权 收集-free, 否则原点权不是 收集-free 的.

这一部分通过精细实现可以做到线性的时空复杂度.

总结

$o = 1$ 的部分可以用充分性证明中的构造简单解决. $o = 2$ 的部分可以用上文中的贪心算法解决.

至此, 我们就对每个用到的关键结论进行了简单的证明, 并用 **线性** 的时空复杂度解决了问题.

致谢

这个问题暂未找到参考资料.

特别感谢曾蕴钦同学分享了他的想法, 并与我讨论本题的做法与加强.

特别感谢唐一文学长提供的较为简单的证明方式.

感谢唐一文学长和曾蕴钦同学参与验题过程.

另: 本题标题意为 "收集操作你的性质太美好", 请勿过度解读.