# 解题报告

周任飞

October 24, 2019

## 1 Logistical Questions

### 1.1 题目大意

给定一棵树，边有长度 $l_i$，点有权值 $w_i$。定义 $\mathrm{dist}(i,j)$ 表示点 $i$ 到点 $j$ 的距离，定义 $\mathrm{cost}(i,j) = \mathrm{dist}(i,j)^{3/2}$。你要选择一个顶点 $u$ 举办比赛，最小化 $\sum_{i=1}^{n} w_i \,\mathrm{cost}(i,u)$（称为“费用”）。

### 1.2 数据范围

$1 \le n \le 2 \times 10^5$，$0 \le w_i \le 10^8$，$1 \le l_i \le 1000$。

你的答案与参考答案的相对或绝对误差不能超过 $10^{-6}$。

### 1.3 解题过程

如果 $\forall i \in [1,n] \quad w_i = 0$，那么选择任意一个顶点，费用为 0。下面假设至少有一个 $w_i \ge 1$。

暂时假设我们不仅可以选择顶点，还可以选择边上任意的一点。具体来说，如果有一条边连接 $A, B$ 且长度为 $l$，则对于所有正实数 $d_1, d_2$ 满足 $d_1 + d_2 = l$，我们可以选择虚拟的点 $C$ 使得 $\mathrm{dist}(A,C) = d_1$ 且 $\mathrm{dist}(B,C) = d_2$。下面将虚拟点和原图的顶点统称为“广义点”，并定义 $P(A,B,x)$ 表示这样一个广义点，它在从 $A$ 到 $B$ 的简单路径上，且与 $A$ 的距离是 $x$。按照这种方法，上面所说的点 $C$ 可以表示为 $P(A,B,d_1)$。

设 $f$ 是定义在广义点上的函数，表示选择这个广义点的费用。定义一条路径 $A \rightsquigarrow B$ 上的 $f$ 为一个关于 $x$ 的函数 $h(x)$，其定义域是 $x \in [0, \mathrm{dist}(A,B)]$；其值 $h(x) = f(P(A,B,x))$。

我们列举如下结论：

(a) 设 $A$ 和 $B$ 是两个顶点，则 $A \rightsquigarrow B$ 上的 $f$ 在其定义域内是严格凹函数。

(b) 不存在两个广义点 $A \neq B$，使它们都是 $f$ 的极小点。其中 $A$ 是 $f$ 的极小点是指

$$\exists \varepsilon > 0 \quad \forall B \quad (\operatorname{dist}(A, B) < \varepsilon) \to (f(B) \geq f(A)).$$

(c) $f$ 的最小点存在且唯一。

下面设 $f$ 的最小点是 $R$。以 $R$ 为根定义广义点之间的祖先后代关系，即 $A$ 是 $B$ 的祖先当且仅当 $\operatorname{dist}(R, A) + \operatorname{dist}(A, B) = \operatorname{dist}(R, B)$，这等价于 $A$ 在简单路径 $R \rightsquigarrow B$ 上。

(d) 如果 $A$ 是 $B$ 的祖先且 $A \neq B$，那么 $f(A) < f(B)$。

(e) 设 $(A, B)$ 是树边。则 $h(x) = f(P(A, B, x))$ 在 $x = 0$ 处有右导数。下面将它记作 $g(A, B)$，称作 $A \to B$ 的梯度。

(f) 设 $A$ 是一个顶点，其邻居为 $B_1, B_2, \ldots, B_k$。可以在 $O(n)$ 的时间复杂度内，对于每一个 $i$ 求出 $A \to B_i$ 的梯度。

(g) 如果 $(A, B)$ 是树边且 $A$ 是 $B$ 祖先，那么 $g(B, A) < 0$。

证明见附录。

**定理 1**：设 $A$ 是一个顶点，它的邻居是 $B_1, B_2, \ldots, B_k$。则：

(1) 如果 $\forall i \in [1, k] \quad g(A, B_i) \geq 0$，那么 $A$ 是 $f$ 的最小点。

(2) 如果对于恰好一个 $i$ 有 $g(A, B_i) < 0$，那么删去顶点 $A$ 后 $B_i$ 和 $R$ 属于同一个连通块。

请注意，删去顶点 $A$ 并不会删除和 $A$ 相邻的边，这将导致边 $(A, B_i)$ 存在但是其一个端点 $A$ 已经被删除（也可能两个端点都已被删除）。

(3) 不可能存在两个 $i$ 满足 $g(A, B_i) < 0$。

证明见附录。

设 $A$ 是树的重心。如果定理 1 的情况 (1) 成立，那么我们就直接得到了答案；如果情况 (2) 成立，那么我们可以删去点 $A$，接着删去不包含 $R$ 的那些连通块。对于剩余的连通块找出所有顶点的重心 $A'$，这一步将不考虑只有一个端点的边造成的影响。对于 $A'$ 进行上述过程……如此一直重复下去，直到某一轮中情况 (1) 成立，或者没有任何剩余顶点。在后一种情况中，一条边将被留下来，而它的两个端点都已经被删除，此时可以断言 $R$ 在这条边上。

因为每一轮过后剩余顶点数都变成原来的至多一半，所以 $\lceil \log_2 n \rceil + 1$ 轮之后这个过程就将结束。再结合结论 (f)，我们可以在 $O(n \log n)$ 的时间复杂度内求出 $R$（如果它恰好是一个顶点）或者求出包含 $R$ 的边。

现在让注意力回到原问题上。如果 $R$ 恰好是一个顶点，那么它就是原问题的答案；否则，设 $R$ 在边 $(A, B)$ 上，那么所有的顶点要么是 $A$ 的后代，要么是 $B$ 的后代。结合结论 (d)，这说明 $A$ 和 $B$ 之一就是原问题的答案，我们可以单独计算它们的答案并加以比较。至此，我们得到了原问题的完整解法，时间复杂度 $O(n \log n)$。

## 1.4 附录

本题中用到的结论 (a) – (g) 以及定理 1 的证明都不难，但需要用到较为繁琐的公式推导，故放在附录中供读者查阅。

**结论 (a) 证明**：设 $h(x) = f(P(A, B, x))$ 是结论中描述的函数。根据定义有

$$h(x) = \sum_{i=1}^{n} w_i \operatorname{dist}(i, P(A, B, x))^{3/2}.$$

下面证明对于任意的 $i$，$t(x) = \operatorname{dist}(i, P(A, B, x))^{3/2}$ 是严格凹函数。

设路径 $A \rightsquigarrow B$ 上距离顶点 $i$ 最近的点是 $C$, $\operatorname{dist}(C, i) = y_0$, $\operatorname{dist}(A, C) = x_0$，则有

$$\operatorname{dist}(i, P(A, B, x)) = |x - x_0| + y_0,$$

进而

$$t(x) = (|x - x_0| + y_0)^{3/2},$$

令 $t_2(x) = t(x + x_0) = (|x| + y_0)^{3/2}$，则只需证明 $t_2(x)$ 在 $\mathbb{R}$ 上是严格凹函数。根据严格凹函数的定义，需要证明

$$\forall x_1 < x_2 \quad \forall x \in (x_1, x_2) \quad t_2(x) < \frac{(x - x_1)t_2(x_1) + (x_2 - x)t_2(x_2)}{x_2 - x_1}.$$

这个命题是说，如果我们在函数的图像上任取两个点，并用一条线段连接它们，那么它们之间的函数图像都严格位于该线段下方。

因为 $t_2(x)$ 是偶函数，不失一般性地，仅考虑两种情况：

1. $0 \le x_1 < x_2$。设 $\hat{t}_2(x)$ 是 $t_2(x)$ 在 $[0, +\infty)$ 上的约束（即定义在 $[0, +\infty)$ 上且与 $t_2(x)$ 具有相同的值的函数），$\hat{t}_2(x)$ 在其定义域内可导，且其导数 $\hat{t}'_2(x) = \frac{3}{2}\sqrt{x + y_0}$ 严格单调递增。由此可以推出 $\hat{t}_2(x)$ 在 $[0, +\infty)$ 上是严格凹函数（这一步的详细证明可以在任何一本微积分教材上找到，这里不再赘述）。再次根据定义，当 $0 \le x_1 < x_2$ 时上面的命题成立。

2. $x_1 < 0 < x_2$。利用上面的几何解释，我们设 $P_1 = (x_1, t_2(x_1))$ 表示 $t_2(x)$ 函数图像与 $x = x_1$ 的交点，同理设 $P_2 = (x_2, t_2(x_2))$、$P_0 = (0, t_2(0))$。因为 $t_2$ 在 $(-\infty, 0]$ 单调递减、在 $[0, +\infty)$ 单调递增，所以 $P_0$ 在线段 $P_1P_2$ 下方，进而线段 $P_1P_0$、$P_0P_2$ 也在线段 $P_1P_2$ 下方。根据前一段中的分析，区间 $(x_1, 0)$ 上的函数图像在线段 $P_1P_0$ 的下方，区间 $(0, x_2)$ 上的函数图像在线段 $P_0P_2$ 的下方，进而它们都在线段 $P_1P_2$ 的下方；又因为 $P_0$ 也在该线段下方，故整一段函数图像 $(x_1, x_2)$ 都在该线段下方，上面的命题成立。

至此，我们证明了 $t(x) = \operatorname{dist}(i, P(A, B, x))^{3/2}$ 是严格凹函数。回到 $h(x)$ 的定义，和式的每一项要么是 0，要么是一个严格凹函数；又因为至少有一个 $w_i > 0$，所以 $h(x)$ 是至少一个严格凹函数的和，它也是一个严格凹函数。 □

**结论 (b) 证明**：用反证法。假设存在两个广义点 $A$ 和 $B$，它们都是 $f$ 的极小点。不失一般性，假设 $f(A) \ge f(B)$。令 $h(x) = f(P(A, B, x))$。根据结

论 (a)，$h(x)$ 是严格凹函数，但根据定义存在 $\varepsilon > 0$ 使 $h(\varepsilon) \geq h(0)$，从而 $h(0) \leq h(\varepsilon) \geq h(\mathrm{dist}(A, B))$，与严格凹函数的定义矛盾。 □

**结论 (c) 证明**：先证存在。每一条边上的 $f$ 是闭区间上的连续函数，故其能取到最小值；而这 $n-1$ 条边上的广义点的并集等于全集，这些最小值的集合有最小值，故 $f$ 能取到最小值，即存在最小点。

再证唯一。根据定义最小点必定是极小点，如果存在两个最小点，就与结论 (b) 矛盾。 □

**结论 (d) 证明**：用反证法，假设 $f(A) \geq f(B)$。设 $h(x) = f(P(R, B, x))$、$\mathrm{dist}(R, A) = x_1$、$\mathrm{dist}(R, B) = x_2$，则有 $h(0) < h(x_1) \geq h(x_2)$，与 $h(x)$ 是严格凹函数相矛盾。 □

**结论 (e) 证明**：根据定义

$$h(x) = \sum_{i=1}^{n} w_i \,\mathrm{dist}(i, P(A, B, x))^{3/2},$$

根据前面的讨论，对于每一个 $i$，都存在 $x_0, y_0$ 使得 $\mathrm{dist}(i, P(A, B, x)) = |x - x_0| + y_0$。设 $t(x) = (|x - x_0| + y_0)^{3/2}$，则在 0 的某个邻域内，要么 $t(x) = (x - x_0 + y_0)^{3/2}$，要么 $t(x) = (x_0 + y_0 - x)^{3/2}$。不论哪种情况，$t(x)$ 在 $x = 0$ 处的右导数都存在，并且容易计算。

$h(x)$ 是若干个这样的 $t(x)$ 的线性组合，所以它在 $x = 0$ 处也有右导数。 □

**结论 (f) 证明**：在上面的讨论中，$t(x)$ 的具体形式取决于 $x_0 = 0$ 是否成立。

$$t'(0+) = \begin{cases} \frac{3}{2}\sqrt{\mathrm{dist}(i, A)}, & \text{if } x_0 = 0, \\ -\frac{3}{2}\sqrt{\mathrm{dist}(i, A)}, & \text{otherwise.} \end{cases}$$

而 $x_0 = 0$ 是否成立取决于，删去顶点 $A$ 后，$i$ 和 $B$ 是否在同一个连通块内（即，$i$ 在以 $A$ 为根 $B$ 的子树内）。进一步推出

$$h'(0+) = \frac{3}{2}\sum_{i=1}^{n} w_i \sqrt{\mathrm{dist}(i, A)}\,(-1)^{\lambda(i, A, B)},$$

其中 $\lambda(i, A, B)$ 表示 $i$ 是否在以 $A$ 为根 $B$ 的子树内，其值为 0 或 1。

删去 $A$ 之后树将剩下 $k$ 个连通块，$B_1, B_2, \ldots, B_k$ 分别属于不同的连通块，令 $S_i$ 表示 $B_i$ 所属的连通块的顶点集。则

$$g(A, B_i) = \frac{3}{2}\sum_{j=1}^{k}(-1)^{[i=j]}\sum_{u \in S_j} w_u \sqrt{\mathrm{dist}(u, A)}.$$

首先从 $A$ 出发运行一次 DFS，对于所有的 $u$ 求出 $\mathrm{dist}(u, A)$，并对于每一个 $S_i$ 求出

$$G_i = \sum_{u \in S_i} w_u \sqrt{\mathrm{dist}(u, A)}.$$

最后依次计算 $g(A,B_i)=\left(\sum_{j=1}^{k}G_j\right)-2G_i$。该算法的时间复杂度是 $O(n)$。$\square$

**结论 (g) 证明**：设 $h(x)=f(P(B,A,x))$，则 $h(x)$ 在定义域内可导且为凹函数，可以推出其导数 $h'(x)$ 在定义域内单调递增。再根据结论 (d) 有 $f(A)<f(B)$，用拉格朗日中值定理得 $\exists\xi\in(0,L)\quad h'(\xi)=\frac{f(A)-f(B)}{L}<0$。根据 $h'(x)$ 单调递增得出 $h'(0+)<0$，即 $g(B,A)<0$。 $\square$

**推论**：如果 $(A,B)$ 是树边且删去 $A$ 后 $R$ 和 $B$ 在同一个连通块中，则 $g(A,B)<0$。

**推论证明**：如果 $R$ 在边 $(A,B)$ 上，则对于 $B'=A,A'=R$ 重复结论 (g) 的证明即可；否则 $B$ 是 $A$ 的祖先，根据结论 (g) 即可得出。 $\square$

**定理 1 证明**：设删去 $A$ 后的连通块分别是 $S_1,S_2,\ldots,S_k$；根据前面的讨论，可以计算出 $G_1,G_2,\ldots,G_k$，使得 $g(A,B_i)=G_S-2G_i$，其中 $G_S=\sum_{j=1}^{k}G_j$。

如果 $A$ 是最小点，那么 $\forall i\in[1,k]\quad g(A,B_i)\geq 0$；如果最小点在 $S_i$ 内，则 $g(A,B_i)<0$。还需证明至多有一个 $i$ 满足 $g(A,B_i)<0$。

事实上，$g(A,B_i)=G_S-2G_i<0\Leftrightarrow G_i>\frac{G_S}{2}$，又因为 $\forall i\in[1,k]\quad G_i\geq 0$，所以至多有一个 $i$ 满足 $g(A,B_i)<0$。

我们已经说明不存在除上述两种情况之外的其他情况，故定理 1 得证。 $\square$

# 2 Cool Slogans

## 2.1 题目大意

给定字符串 $S$。求最长的字符串序列 $s_1, s_2, \ldots, s_k$ 的长度，满足

1. $\forall i \in [1, k]$，$s_i$ 是 $S$ 的非空子串。
2. $\forall i \in [2, k]$，$s_{i-1}$ 在 $s_i$ 中作为子串出现了至少两次。多次出现之间可以重叠。

## 2.2 数据范围

$1 \le |S| \le 2 \times 10^5$，$S$ 仅包含小写英文字母。

## 2.3 记号与约定

解题过程中需要用到后缀自动机（简称 SAM），故在本文中假设读者已经能熟练运用该数据结构。

设 $A$ 是一个字符串，将其中的字符从左到右按照 1 到 $n$ 编号，令 $A(i)$ 表示它的第 $i$ 个字符。定义 $A[l, r]$ 表示字符串的一个子串，它由编号在区间 $[l, r]$ 内的字符组成；$l$ 和 $r$ 分别称作其左端点和右端点。

在不引起混淆的前提下，用“母串”表示原问题中输入的字符串（即 $S$），也将直接使用“子串”来表示母串的子串。

定义一个子串 $A$ 的 right 集合为其在母串中所有出现位置的右端点的集合。严格来说，

$$\mathrm{right}(A) = \{r \in [|A|, n] \mid S[r - |A| + 1, r] = A\}$$

SAM 结点是一个字符串集合，其元素的长度取到某一个区间内的所有值，并且所有元素都是最长的元素的后缀。可以用一个三元组 $(L, R, E)$ 表示一个 SAM 结点，表示 $\{S[i, E] \mid i \in [L, R]\}$。SAM 结点的各个元素拥有相同的 right 集合，也称为这个结点的 right 集合。

SAM 的各个结点之间按照后缀关系形成的树结构，称作反向前缀树。在不引起混淆的前提下，也简称作“树”。

## 2.4 解题过程

对母串建 SAM，并使用可持久化线段树合并求出每一个结点的 right 集合。

**定理 1**：如果限制序列中 $s_{i-1}$ 必须是 $s_i$ 的后缀，答案不会改变。

**定理 1 证明**：暂时将这一条限制称为“规则”。首先，符合规则的序列一定是原问题中合法的序列，所以答案不会变大，下面证明答案不会变小。

如果原问题的最优解 $s_1, s_2, \ldots, s_k$ 中某个 $s_{i-1}$ 不是 $s_i$ 的后缀，那么违反规则的 $i$ 中存在一个最小的。令 $s_i'$ 是 $s_i$ 的一个最长的前缀，满足 $s_{i-1}$ 是其后缀，

则容易发现 $s_{i-1}$ 在 $s_i'$ 中出现了至少两次，$s_i'$ 在 $s_{i+1}$（如果存在）中也出现了至少两次。用 $s_i'$ 代替 $s_i$ 后序列同样满足题目中的条件，且这个序列中要么不存在违反规则的 $i$，要么最小的违反规则的 $i$ 比原序列中大。如此迭代下去可以得到一个相同长度的符合规则的序列。 □

在本文的剩余部分，我们将这个约束视为原问题的一部分。

设 $A$ 是某个子串，定义 $f(A)$ 表示最长的以 $A$ 为最后一项的序列长度。那么

$$f(A) = 1 + \max\{f(B)\}$$

其中 $B$ 是 $A$ 的后缀且在 $A$ 中出现至少两次。如果这个条件成立，就称字符串 $B$ 能转移到 $A$。

**定理 2**：在 SAM 结点 $X$ 中，$A$ 是其最长的元素，$B$ 是任意其他元素。则对于另一个 SAM 结点 $Y$ 的最长元素 $C$，如果 $B$ 能转移到 $C$，则 $A$ 也能转移到 $C$。

**定理 2 证明**：因为 $B$ 能转移到 $C$，所以 $X$ 和 $Y$ 的 right 集合有交。设 $X = (L_1, R_1, E), Y = (L_2, R_2, E)$。令 $E'$ 表示 $X$ 的 right 集合中 $E$ 的前驱，则需要证明

$$(E' + R_1 - E \geq L_2) \rightarrow (E' + L_1 - E \geq L_2).$$

用反证法。假设

$$\begin{cases} E' + R_1 - E \geq L_2, \\ E' + L_1 - E < L_2, \end{cases}$$

令 $L_1' = E' - E + L_1$、$R_1' = E' - E + R_1$，那么 $X = (L_1', R_1', E')$，上式变为 $L_1' < L_2 \leq R_1'$。因为 $S[L_2 - 1, E']$ 和 $T = S[L_2, E']$ 在同一个 SAM 结点 $X$ 中，所以它们有相同的 right 集合，即后者在母串中的每一次出现时，左端点都不是第一个字符，且左端点左侧相邻的字符全都相同。此外，由 $T' = S[L_2, E]$ 是 SAM 结点 $Y$ 的最长元素，可知其在母串中的每一次出现时，要么出现左端点是第一个字符的情况，要么左端点左侧的字符不尽相同。但是 $T$ 是 $T'$ 的子串，这与刚才获知的信息矛盾。 □

容易发现在一个 SAM 结点中，其最长的元素 $f$ 值最大。结合定理 2，这说明我们只需要关心每一个 SAM 结点的最长元素即可。下面兼用一个 SAM 结点的记号来指代其最长的元素。特别地，记 $f(X)$（$X$ 是一个 SAM 结点）表示该结点最长元素的 $f$ 值。

我们列举如下结论：

(a) 如果树上 $A$ 是 $B$ 的父亲，那么 $f(A) \leq f(B)$。

(b) 如果树上 $A$ 是 $B$ 的父亲，$C$ 是 $B$ 的后代，则 $B$ 能转移到 $C$ 蕴含 $A$ 能转移到 $C$。

(c) 如果树上 $A$ 是 $B$ 的祖先，$C$ 是 $B$ 的孩子，则 $A$ 能转移到 $B$ 蕴含 $A$ 能转移到 $C$。

(d) 树上能转移到 $A$ 的那些结点形成一条从根节点到 $A$ 的某个祖先的链。

证明详见附录。

结论 (a)、(d) 共同指出，我们只需要对于每一个 $A$，求出深度最大的能转移到 $A$ 的顶点，就可以对于每一个 $A$ 求出 $f(A)$。设深度最大的能转移到 $A$ 的顶点为 $g(A)$。

按照深度从小到大的顺序对于每一个点 $A$ 计算 $g(A)$。假设 $A$ 的父亲是 $B$，那么有 $g(A)$ 是 $g(B)$ 的后代（可以等于 $g(B)$）。为了计算 $g(A)$，从 $g(B)$ 的孩子开始枚举从 $g(B)$ 到 $A$ 的链上的各个结点，并逐个检测它是否能转移到 $A$。一旦结果为“否”，就将上一个结点记为 $g(A)$ 并立刻停止枚举。在整个过程中，每一个结点最多作为 $A$ 导致一次结果为“否”的检测，最多作为上方的结点导致一次结果为“是”的检测，因此总共的检测次数不超过 SAM 结点数的两倍，即 $4n$。

为了检测 $A$ 是否能转移到 $B$，我们任取 $\mathrm{right}(B)$ 的任意一个元素 $E$，找到 $\mathrm{right}(A)$ 中其前驱 $E'$，并判断 $A$ 以 $E'$ 为右端点的出现是否完全含在 $B$ 以 $E$ 为右端点的出现之内。其中找前驱需要用到起初建立的线段树结构，一次检测的时间复杂度是 $O(\log n)$。

计算出所有 $g(A)$ 之后，再按深度从小到大的顺序计算每一个点的 $f(A)$，这时令 $f(A) = f(g(A)) + 1$ 即可。边界条件是根节点的 $f$ 值为 0。

至此，我们得到原问题的完整解法，时间复杂度 $O(n \log n)$。

## 2.5 附录

**结论 (a) 证明**：$A$ 是 $B$ 的父亲可以推出 $A$ 是 $B$ 的后缀，在以 $A$ 为最后一项的最长序列中将最后一项替换为 $B$ 并不会违反条件，从而 $f(A) \le f(B)$。□

**结论 (b) 证明**：如果 $B$ 能转移到 $C$，那么 $B$ 在 $C$ 中出现至少两次，而 $A$ 是 $B$ 的子串，所以 $A$ 也在 $C$ 中出现至少两次。□

**结论 (c) 证明**：如果 $A$ 能转移到 $B$，那么 $A$ 在 $B$ 中出现至少两次，而 $B$ 是 $C$ 的子串，所以 $A$ 也在 $C$ 中出现至少两次。□

**结论 (d) 证明**：根据转移的定义，能转移到 $A$ 的结点必定是 $A$ 的后缀，即树上 $A$ 的祖先；再根据结论 (b) 即可得出。□

# 3 Two Faced Edges

## 3.1 题目大意

给定 $n$ 个顶点、$m$ 条边的有向图，对于每一条边回答，将其反向之后图中 SCC（强连通分量）的个数是否会发生变化。

将一条边 $A \to B$ 反向的意思是，删除这条边，并加入边 $B \to A$。

## 3.2 数据范围

$2 \le n \le 1000$，$1 \le m \le 2 \times 10^5$，给定的图没有重边或自环。

## 3.3 解题过程

**定理 1**：将一条边 $A \to B$ 反向后 SCC 数量是否发生变化，仅取决于下面两个问题的答案：

1. $A$ 是否可以不经过这条边到达 $B$。
2. $B$ 是否可以到达 $A$。

证明详见附录。

问题 2 是容易解决的。只需要从每一个顶点出发分别运行一次 DFS，求出它能到达哪些顶点，再对每一条边逐一回答即可。时间复杂度 $O(nm)$。

接下来，我们将用 $O(m)$ 的时间，对所有以 $P$ 为起点的边回答问题 1，其中 $P$ 是任意顶点。

假设以 $P$ 为起点的边共有 $k$ 条，终点分别是 $Q_1, Q_2, \ldots, Q_k$。对于每一个 $Q_i$，我们需要判断，是否有一条简单路径 $P \to Q_j \rightsquigarrow Q_i$，其中 $j \neq i$。对于 $P$ 能到达的顶点 $X$，定义 $\mathrm{Min}(X)$ 是最小的 $j$，满足存在简单路径 $P \to Q_j \rightsquigarrow X$，这等价于存在一条不经过 $P$ 的路径 $Q_j \rightsquigarrow X$。同理可以定义 $\mathrm{Max}(X)$ 是满足条件的最大的 $j$。

从小到大枚举每一个 $i$，在不允许经过 $P$ 的前提下，以 $Q_i$ 为起点运行一次 DFS，把所有未被标记过的顶点标记为“已访问”。对于所有在这一轮新标记的顶点 $X$，令 $\mathrm{Min}(X) = i$。

这样我们就对于所有的 $i$ 求出了 $\mathrm{Min}(Q_i)$。再从大到小枚举每一个 $i$，执行相同的算法，我们就对于所有的 $i$ 求出了 $\mathrm{Max}(Q_i)$。再检查是否有 $\mathrm{Min}(Q_i) = \mathrm{Max}(Q_i) = i$，就可以对 $P \to Q_i$ 这条边回答问题 1。

在上面的过程中，每个顶点只会被标记一次，进而每一条边也只会在 DFS 中被访问一次，时间复杂度是 $O(m)$。结合前面的转化，我们就以 $O(nm)$ 的时间复杂度解决了原问题。

## 3.4 附录

为方便下文的叙述，当 $G$ 上存在一条路径以 $X$ 为起点、以 $Y$ 为终点，就称（$G$ 上）$X$ 可达 $Y$，记作 $X \rightsquigarrow Y$。如果 $X$ 和 $Y$ 相互可达，记作 $X \leftrightsquigarrow Y$。

为了证明定理 1，我们先提出几个引理。

**引理 1**：设有向图 $G$ 加入边 $A \to B$ 后形成图 $G'$，有 $\mathrm{Count}(G') \le \mathrm{Count}(G)$，其中 $\mathrm{Count}(G)$ 表示图 $G$ 中 SCC 的数量。

**引理 1 证明**：设 $G$ 的 SCC 有 $p$ 个，分别是 $S_1, S_2, \ldots, S_p$；$G'$ 的 SCC 有 $q$ 个，分别是 $T_1, T_2, \ldots, T_q$。如果 $G$ 上 $X \rightsquigarrow Y$，那么 $G'$ 上 $X \rightsquigarrow Y$；进一步地，如果 $G$ 上 $X \leftrightsquigarrow Y$，那么 $G'$ 上也有 $X \leftrightsquigarrow Y$。

对于每一个 $T_i$，取出其中任意一个点，设为 $K_i$，则存在恰好一个 $j$ 使 $K_i \in S_j$。假设 $q > p$，根据鸽巢原理，一定存在 $i \ne i'$ 使得 $K_i$ 和 $K_{i'}$ 属于同一个 $S_j$，即存在 $j$ 使 $(K_i \in S_j) \land (K_{i'} \in S_j)$。这说明 $G$ 上 $K_i \leftrightsquigarrow K_{i'}$，但 $G'$ 上不是。根据上面的推理，这是不可能的。 $\square$

**引理 2**：$\mathrm{Count}(G') < \mathrm{Count}(G)$ 的充分必要条件是，存在顶点 $X$ 和 $Y$，它们在 $G'$ 上相互可达，但在 $G$ 上不是。

**引理 2 证明**：先证充分性。设 $X \in S_a$，$Y \in S_b$，$X, Y \in T_c$，对于除 $T_c$ 外的每一个 $T_i$，取出其中任意一个点，设为 $K_i$，则存在恰好一个 $j$ 使 $K_i \in S_j$，这个 $j$ 必定满足 $(j \ne a) \land (j \ne b)$。假设 $(q-1) > (p-2)$，使用和引理 1 相同的方法可以推出矛盾，于是 $q \le p-1$。

再证必要性。对于每一个 $S_i$ 取出其中任意一个点，设为 $K'_i$，则存在恰好一个 $j$ 使 $K'_i \in T_j$。因为 $p > q$，根据鸽巢原理，存在 $i \ne i'$ 使 $K'_i$ 和 $K'_{i'}$ 属于同一个 $T_j$，则 $X = K'_i$ 和 $Y = K'_{i'}$ 在 $G'$ 上相互可达，但在 $G$ 上不是。 $\square$

下面称在 $G'$ 上互相可达、在 $G$ 上不互相可达的无序点对 $(X, Y)$ 是**巧克力点对**。

**推论 1**：如果存在巧克力点对，那么 $\mathrm{Count}(G') < \mathrm{Count}(G)$；否则，$\mathrm{Count}(G') = \mathrm{Count}(G)$。

**推论 1 证明**：由引理 1 和引理 2 直接推出。 $\square$

**引理 3**：加入边 $A \to B$ 后存在巧克力点对的充分必要条件是，$A \not\rightsquigarrow B \land B \rightsquigarrow A$。

**引理 3 证明**：分三种情况讨论。

1. $A \rightsquigarrow B$。加入边 $A \to B$ 不会改变任何点对的可达性，所以不存在巧克力点对。
2. $B \not\rightsquigarrow A$。设 $G'$ 上 $X \rightsquigarrow Y$，但 $G$ 上 $X \not\rightsquigarrow Y$。那么 $G'$ 存在路径 $X \rightsquigarrow A \to B \rightsquigarrow Y$。如果 $G'$ 上 $Y \rightsquigarrow X$，则推出 $G'$ 上 $B \rightsquigarrow A$，进一步推出 $G$ 上 $B \rightsquigarrow A$，矛盾，所以 $G'$ 上 $Y \not\rightsquigarrow X$。因此不存在巧克力点对。
3. $A \not\rightsquigarrow B \land B \rightsquigarrow A$。$(A, B)$ 是一对巧克力点对。 $\square$

**推论 2**：如果 $A \not\rightsquigarrow B \land B \rightsquigarrow A$，那么 $\mathrm{Count}(G') < \mathrm{Count}(G)$；否则，$\mathrm{Count}(G') = \mathrm{Count}(G)$。

**推论 2 证明**：由引理 3 和推论 1 直接推出。 $\square$

现在我们关注反转一条边的过程。设图 $G$ 加入边 $A \to B$ 形成图 $G'$，加入边 $B \to A$ 形成图 $G''$。这样 $G''$ 是 $G'$ 反转 $A \to B$ 这条边形成的图。

**引理 4**：$\text{Count}(G') = \text{Count}(G)$ 和 $\text{Count}(G'') = \text{Count}(G)$ 至少有一个成立。

**引理 4 证明**：$A \not\leadsto B \wedge B \leadsto A$ 和 $A \leadsto B \wedge B \not\leadsto A$ 不可能同时成立。 $\square$

**推论 3**：$\text{Count}(G') \neq \text{Count}(G'')$ 的充分必要条件是 $(A \not\leadsto B \wedge B \leadsto A) \vee (A \leadsto B \wedge B \not\leadsto A)$。

**推论 3 证明**：根据引理 4，$\text{Count}(G') \neq \text{Count}(G'')$ 当且仅当 $\text{Count}(G') < \text{Count}(G)$ 或 $\text{Count}(G'') < \text{Count}(G)$。再根据推论 2 即可得到这个条件。 $\square$

推论 3 不仅说明了定理 1 是正确的，还指出了该如何从两个子问题的答案得到原问题的答案，至此定理 1 得证。