Solution

ExfJoe

福建省长乐第一中学

March 14, 2017

Outline

1. 编码
2. 哈密顿回路
3. 旅行

编码

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可
- 100 分：上述做法问题在于边数太多，因此我们考虑先枚举问号，然后把所有可能串建成一棵 Trie

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可
- 100 分：上述做法问题在于边数太多，因此我们考虑先枚举问号，然后把所有可能串建成一棵 Trie
- 设 $a_{i,j}$ 表示 $x_i = j$ 时 s_i 在 Trie 中编号

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可
- 100 分：上述做法问题在于边数太多，因此我们考虑先枚举问号，然后把所有可能串建成一棵 Trie
- 设 $a_{i,j}$ 表示 $x_i = j$ 时 s_i 在 Trie 中编号
- 对于 Trie 中每个结点 i，设 y_i 表示是否存在一个被选的点是 i 所代表的串的前缀

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可
- 100 分：上述做法问题在于边数太多，因此我们考虑先枚举问号，然后把所有可能串建成一棵 Trie
- 设 $a_{i,j}$ 表示 $x_i = j$ 时 s_i 在 Trie 中编号
- 对于 Trie 中每个结点 i，设 y_i 表示是否存在一个被选的点是 i 所代表的串的前缀
- 接着我们可以得到若干逻辑关系，我们可以将 $(x_i = j)$ 与 $(y_{a_{i,j}} = 1)$ 连边，然后 Trie 上的父亲与儿子互相连边。注意处理一下多个 s_i 在同一结点的情况，剩下的依然是跑 2-SAT

编码

- 50 分：考虑将第 i 个串的问号看做一个布尔变量 x_i
- 对于两个串 i, j，我们枚举 x_i, x_j 分别填了什么，若发生了冲突 (不满足前缀编码性质)，则我们可以得到对应的逻辑关系，比如 $(x_i = 0) \Rightarrow (x_j = 1)$
- 这是一个 2-SAT 模型，跑 Tarjan 求解即可
- 100 分：上述做法问题在于边数太多，因此我们考虑先枚举问号，然后把所有可能串建成一棵 Trie
- 设 $a_{i,j}$ 表示 $x_i = j$ 时 s_i 在 Trie 中编号
- 对于 Trie 中每个结点 i，设 y_i 表示是否存在一个被选的点是 i 所代表的串的前缀
- 接着我们可以得到若干逻辑关系，我们可以将 $(x_i = j)$ 与 $(y_{a_{i,j}} = 1)$ 连边，然后 Trie 上的父亲与儿子互相连边。注意处理一下多个 s_i 在同一结点的情况，剩下的依然是跑 2-SAT
- 主要思想就是利用 Trie 来优化 2-SAT 的连边

Outline

哈密顿回路

哈密顿回路

- 由于点数不大，因此我们考虑枚举起点 u 然后做 meet in the middle

哈密顿回路

- 由于点数不大，因此我们考虑枚举起点 u 然后做 meet in the middle
- 将 u 当做起点向后搜 $\frac{n}{2}$ 步，再当做终点向前搜 $\frac{n}{2}$ 步

哈密顿回路

- 由于点数不大，因此我们考虑枚举起点 u 然后做 meet in the middle
- 将 u 当做起点向后搜 $\frac{n}{2}$ 步，再当做终点向前搜 $\frac{n}{2}$ 步
- 记录路径上点的出现情况，利用 map 查找路径是否存在

哈密顿回路

- 由于点数不大，因此我们考虑枚举起点 u 然后做 meet in the middle
- 将 u 当做起点向后搜 $\frac{n}{2}$ 步，再当做终点向前搜 $\frac{n}{2}$ 步
- 记录路径上点的出现情况，利用 map 查找路径是否存在
- 由于这是个回路，所以其实不用枚举起点，从 1 开始即可

Outline

旅行

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i, a, b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i,a,b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i, a, b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组
- 然而 $|f_i|$ 大小仍然是平方级别，我们考虑若 $(a, b), (c, d)$，$a \leq c, b \leq d$，则 (c, d) 是没用的

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i, a, b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组
- 然而 $|f_i|$ 大小仍然是平方级别，我们考虑若 $(a, b), (c, d)$，$a \leq c, b \leq d$，则 (c, d) 是没用的
- 因此对于 i 的儿子 j, k，$|f_i| \leq 2min(|f_j|, |f_k|)$.

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i, a, b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组
- 然而 $|f_i|$ 大小仍然是平方级别，我们考虑若 $(a, b), (c, d)$，$a \leq c, b \leq d$，则 (c, d) 是没用的
- 因此对于 i 的儿子 j, k，$|f_i| \leq 2min(|f_j|, |f_k|)$.
- 因为若先选择 f_j 中的 (a, b)，则接下来会选择满足条件中 d 最小的 (c, d)

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i,a,b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组
- 然而 $|f_i|$ 大小仍然是平方级别，我们考虑若 $(a,b),(c,d)$，$a \leq c, b \leq d$，则 (c,d) 是没用的
- 因此对于 i 的儿子 j,k，$|f_i| \leq 2min(|f_j|,|f_k|)$.
- 因为若先选择 f_j 中的 (a,b)，则接下来会选择满足条件中 d 最小的 (c,d)
- $\sum |f_i|$ 为 $O(nlogn)$ 级别。将两个儿子的 f 排序后 two points 合并即可

旅行

- 首先容易想到二分答案，因此现在的问题是选取一个叶子遍历顺序，使得中间叶子间距离都 $\leq mid$
- 由于每条树边最多经过两次，所以这是一个 dfs
- $f(i, a, b)$ 表示进入 i 子树，第一天花费为 a，最后一天花费为 b 的情况下，是否能够让中间叶子满足条件
- 由于空间问题，所以将 f 存成一堆二元组
- 然而 $|f_i|$ 大小仍然是平方级别，我们考虑若 $(a, b), (c, d)$，$a \leq c, b \leq d$，则 (c, d) 是没用的
- 因此对于 i 的儿子 j, k，$|f_i| \leq 2min(|f_j|, |f_k|)$.
- 因为若先选择 f_j 中的 (a, b)，则接下来会选择满足条件中 d 最小的 (c, d)
- $\sum |f_i|$ 为 $O(nlogn)$ 级别。将两个儿子的 f 排序后 two points 合并即可
- 时间复杂度 $O(n \log^2 n)$