追忆

Itst, SnowySummer

2025 年 3 月 1 日

问题

给定一个 n 个点 m 条边的有向图 G，结点由 1 至 n 编号。第 i $(1 \le i \le m)$ 条边从 u_i 指向 v_i，保证 $u_i < v_i$。节点 j $(1 \le j \le n)$ 有两个权值 a_j, b_j，保证 $[a_1, \dots, a_n]$ 与 $[b_1, \dots, b_n]$ 均是 $1 \sim n$ 的排列。

你需要进行 q 次操作。操作有以下三种：

- $1\ x\ y$：交换 a_x 和 a_y；
- $2\ x\ y$：交换 b_x 和 b_y；
- $3\ x\ l\ r$：你需要输出满足以下两个条件的点 y 中 b_y 的最大值，若不存在满足条件的点则输出 0：
  1. $l \le a_y \le r$。
  2. 图 G 中存在一条 x 到 y 的有向路径。

数据范围：$1 \le n, q \le 10^5$，$1 \le m \le 2 \times 10^5$。

- 每次询问时，通过一次 BFS 求出所有 x 能到达的点，然后依次判断。
- 时间复杂度为 $O(mq)$。
- 可以预处理每个点能到达的所有点，询问时直接枚举。
- 时间复杂度为 $O(nq)$，可以通过测试点 $1 \sim 5$。

- 判断可达性是困难的，较快的方式只有 Bitset。
- 由于 a 不变，因此对于 $l \le a_i \le r$ 的限制，也可以使用 Bitset 维护。
- 具体地，对于 $1 \le r \le n$，维护 $A[r] = \{i \mid a_i \in [1, r]\}$，
- 则 $l \le a_i \le r$ 的点集为 $(A[l-1] \operatorname{xor} A[r])$。
- 将可达性与权值对应的 Bitset 取交后即为所有满足要求的点。
- 由于 b 不变，可以将所有点按照 b 的大小重标号，然后求出 lowbit 即可。
- 时间复杂度为 $O(n(m+q)/w)$，可以通过测试点 6。

- 由于 a 可能会变，因此不能直接静态地维护前缀 Bitset。
- 考虑使用数据结构维护 a：
  - 若使用带修莫队，则时间复杂度为 $O(qn^{2/3})$。
  - 若使用分块，则时间复杂度为 $O(q\sqrt{n})$。
- 依旧按照 b 的大小重标号后求出 lowbit 即可。
- 时间复杂度为 $O(n(m+q)/w+q\sqrt{n})$，可以通过测试点 $6\sim 9$。

- 只需要计算所有可达到的点，如果没有修改可以直接拓扑排序解决。
- 考虑将询问按时间分块：
  - 对于块内未修改的点，直接一次拓扑排序计算；
  - 对于块内修改过的点，对于每个询问都枚举一遍计算。
- 时间复杂度为 $O(q\sqrt{n})$，可以通过测试点 $10 \sim 12$。

- 考虑沿用特殊性质 AB 的算法，先求出可行点集。
- 将 $1 \sim n$ 分为 $\log n$ 大小的块，每块内可以预处理出所有 $2^{\log n} = n$ 种情况的答案，询问时直接计算每块的答案即可。
- 时间复杂度为 $O(n(m+q)/w + n(n+q)/\log n)$，可以通过测试点 6、$10 \sim 16$。

- 从大到小依次考虑每个 b 对询问的影响。
- 如果没有修改，可以直接维护出所有包含当前权值的询问集合，然后与还未计算出答案的询问集合取交，此时得到的集合即为所有答案为当前权值的询问，可以依次枚举赋值。
- 考虑将询问按时间分块：
  - 对于块内未修改的点，按照上述方法计算；
  - 对于块内修改过的点，对于每个询问都枚举一遍计算。
- 时间复杂度为 $O(q(n+m)/w)$，可以通过测试点 $17, 18$。

- 在特殊性质 D 的基础上加入权值 a 的影响。
- 考虑预处理出所有包含某一权值 a 的询问。
- 将询问按时间分块后，每块内的 a 段数有限，直接维护前缀 Bitset 即可。
- 时间复杂度为 $O(q(n+m)/w)$，可以通过测试点 $1 \sim 25$。