

战略游戏 (game) 解题报告

quailty

2018 年 5 月 15 日

战略游戏 (game.c/cpp/pas)

- 给一张 n 个点 m 条边的无向连通图，有 q 次询问，每次给出一个点集 S ，你要求出有多少个点 x 使得删去 x 之后 S 不连通。

战略游戏 (game.c/cpp/pas)

- 给一张 n 个点 m 条边的无向连通图，有 q 次询问，每次给出一个点集 S ，你要求出有多少个点 x 使得删去 x 之后 S 不连通。
- 对于 100% 的数据， $T \leq 10$ ， $2 \leq n \leq 10^5$ ， $n - 1 \leq m \leq 2 \times 10^5$ ， $1 \leq q \leq 10^5$ 。

战略游戏 (game.c/cpp/pas)

- 给一张 n 个点 m 条边的无向连通图，有 q 次询问，每次给出一个点集 S ，你要求出有多少个点 x 使得删去 x 之后 S 不连通。
- 对于 100% 的数据， $T \leq 10$ ， $2 \leq n \leq 10^5$ ， $n - 1 \leq m \leq 2 \times 10^5$ ， $1 \leq q \leq 10^5$ 。
- Shortest judge solution: 2279 bytes

30 分做法

- 对于第 1 个测试点, $\sum |S| \leq 20$ 。

30 分做法

- 对于第 1 个测试点, $\sum |S| \leq 20$ 。
- 枚举删点之后重新建图, 并查集维护连通性, 对于每个询问检查 S 是否连通。

30 分做法

- 对于第 1 个测试点, $\sum |S| \leq 20$ 。
- 枚举删点之后重新建图, 并查集维护连通性, 对于每个询问检查 S 是否连通。
- 复杂度 $O(nm\alpha(n) + n\alpha(n) \sum |S|)$, 但是太慢了。

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，
- 递归处理完 $[l, m]$ 之后撤销掉加入 $[m + 1, r]$ 的点的操作，

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，
- 递归处理完 $[l, m]$ 之后撤销掉加入 $[m + 1, r]$ 的点的操作，
- 再加入 $[l, m]$ 的点，然后递归处理 $[m + 1, r]$ ，

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，
- 递归处理完 $[l, m]$ 之后撤销掉加入 $[m + 1, r]$ 的点的操作，
- 再加入 $[l, m]$ 的点，然后递归处理 $[m + 1, r]$ ，
- 递归处理完 $[m + 1, r]$ 之后撤销掉加入 $[l, m]$ 的点的操作。

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，
- 递归处理完 $[l, m]$ 之后撤销掉加入 $[m + 1, r]$ 的点的操作，
- 再加入 $[l, m]$ 的点，然后递归处理 $[m + 1, r]$ ，
- 递归处理完 $[m + 1, r]$ 之后撤销掉加入 $[l, m]$ 的点的操作。
- 只涉及到加边和撤销操作，带撤销并查集维护连通性。

30 分做法

- 瓶颈在于枚举删点之后重新建图，利用分治优化。
- 假设现在还有 $[l, r]$ 区间的点没有加入图中，
- 先加入 $[m + 1, r]$ 的点，然后递归处理 $[l, m]$ ，
- 递归处理完 $[l, m]$ 之后撤销掉加入 $[m + 1, r]$ 的点的操作，
- 再加入 $[l, m]$ 的点，然后递归处理 $[m + 1, r]$ ，
- 递归处理完 $[m + 1, r]$ 之后撤销掉加入 $[l, m]$ 的点的操作。
- 只涉及到加边和撤销操作，带撤销并查集维护连通性。
- 复杂度 $O(m \log^2 n + n \log n \sum |S|)$ ，期望得分 30 分。

45 分做法

- 对于第 2 个测试点, $|S| = 2$ 。

45 分做法

- 对于第 2 个测试点, $|S| = 2$ 。
- $\sum |S|$ 变得非常大, 看上去前一个做法已经没救了, 先冷静分析一下。

45 分做法

- 对于第 2 个测试点, $|S| = 2$ 。
- $\sum |S|$ 变得非常大, 看上去前一个做法已经没救了, 先冷静分析一下。
- 记删去的点是 x , 那么 x 一定是原图的割点, 否则删去 x 不会使 S 内任意一对点不连通。

45 分做法

- 对于第 2 个测试点, $|S| = 2$ 。
- $\sum |S|$ 变得非常大, 看上去前一个做法已经没救了, 先冷静分析一下。
- 记删去的点是 x , 那么 x 一定是原图的割点, 否则删去 x 不会使 S 内任意一对点不连通。
- 看上去要考虑点双连通分量。

45 分做法

- 如果能类似边双连通分量缩点得到一个树结构，就可以在这棵树上做路径询问。

45 分做法

- 如果能类似边双连通分量缩点得到一个树结构，就可以在这棵树上做路径询问。
- 利用 Tarjan 算法进行求出所有点双连通分量，对每个点双连通分量新建一个点，与所有属于该分量的原图中的点连一条边。

45 分做法

- 如果能类似边双连通分量缩点得到一个树结构，就可以在这棵树上做路径询问。
- 利用 Tarjan 算法进行求出所有点双连通分量，对每个点双连通分量新建一个点，与所有属于该分量的原图中的点连一条边。
- 不难发现这是一棵树，并且在这棵树上删一个点对连通性的影响与在原图中删一个点完全相同。

45 分做法

- 如果能类似边双连通分量缩点得到一个树结构，就可以在这棵树上做路径询问。
- 利用 Tarjan 算法进行求出所有点双连通分量，对每个点双连通分量新建一个点，与所有属于该分量的原图中的点连一条边。
- 不难发现这是一棵树，并且在这棵树上删一个点对连通性的影响与在原图中删一个点完全相同。
- 实际上可以把非割点缩入所属的点双连通分量内，也不会影响结果。

45 分做法

- 现在只需静态查询树上两点之间的简单路径中有多少个点对应原图中的点。

45 分做法

- 现在只需静态查询树上两点之间的简单路径中有多少个点对应原图中的点。
- 方法非常多，这里就不一一列举了，期望得分 45 分。

75 分做法

- “前两个做法都想到了，但还是没想到如何解决整个题。”

75 分做法

- “前两个做法都想到了，但还是没想到如何解决整个题。”
- 不妨先把两份代码都写了然后做下一题，期望得分 75 分。

100 分做法

- 对于第 3 个测试点, $\sum |S| \leq 2 \times 10^5$ 。

100 分做法

- 对于第 3 个测试点, $\sum |S| \leq 2 \times 10^5$ 。
- 仍然考虑点双连通分量缩点, 现在是给一个点集 S , 要查询 S 内任意两点之间的简单路径的并集中有多少个点对应原图中的点。

100 分做法

- 对于第 3 个测试点, $\sum |S| \leq 2 \times 10^5$ 。
- 仍然考虑点双连通分量缩点, 现在是给一个点集 S , 要查询 S 内任意两点之间的简单路径的并集中有多少个点对应原图中的点。
- 虚树, 去年考过的。

100 分做法

- 对于第 3 个测试点, $\sum |S| \leq 2 \times 10^5$ 。
- 仍然考虑点双连通分量缩点, 现在是给一个点集 S , 要查询 S 内任意两点之间的简单路径的并集中有多少个点对应原图中的点。
- 虚树, 去年考过的。
- 每加入一条虚树边 $(u, v) (dep_u < dep_v)$, 把 u 到 v 这一段路径 (不含 u) 的贡献算上, 最后补上虚树根的贡献。

100 分做法

- 对于第 3 个测试点, $\sum |S| \leq 2 \times 10^5$ 。
- 仍然考虑点双连通分量缩点, 现在是给一个点集 S , 要查询 S 内任意两点之间的简单路径的并集中有多少个点对应原图中的点。
- 虚树, 去年考过的。
- 每加入一条虚树边 $(u, v) (dep_u < dep_v)$, 把 u 到 v 这一段路径 (不含 u) 的贡献算上, 最后补上虚树根的贡献。
- 复杂度 $O(m + n \log n + \sum |S| \log \sum |S|)$, 期望得分 100 分。

Thank you!