

BIT-图计算-复赛技术报告

我们的总体的思路是用二分的方法求 $k\text{-max truss}$ 。

针对 k 值来进行二分，维护一个 L, R 的上下界， L 初始值为1， R 初始值为整个图上的最大点的 $degree$ 。

令 $mid = (L + R)/2$ ，try 之前先保存上次成功的子图，如果尝试求解 $mid\text{-truss}$ 失败，则回滚到上次成功的子图，且让 $mid = R$ ；否则，让 $L = mid$ ，在当前成功子图的基础上继续求解 $k\text{-truss}$ 。

每次 try 一个 k 值时，我们都需要将当前子图中， $support$ 值小于 k 的边全部迭代的删除掉，直到稳定为没有边可以删除为止。

计算 $support$ 的过程中，我们按边在 GPU 上并行求解，针对 (u, v) 这条边，我们求 u 和 v 的邻居的交集，复杂度为 $O(\deg(u) + \deg(v))$ 。求出 (u, v) 这条边的 $support$ 的值之后，我们二分的查找 (v, u) 这条反向边的在邻接表中的位置，并为它赋值。