最小环

问题

给出一个图，问其中的有 n 个节点构成的边权和最小的环 $(n \geq 3)$ 是多大。

图的最小环也称围长。

暴力解法

设 u 和 v 之间有一条边长为 w 的边， $dis(u,v)$ 表示删除 u 和 v 之间的连边之后， u 和 v 之间的最短路。

那么最小环是 $dis(u,v)+w$ 。

总时间复杂度 $O(n^2m)$ 。

Dijkstra

枚举所有边，每一次求删除一条边之后对这条边的起点跑一次 Dijkstra，道理同上。

时间复杂度 $O(m(n+m)\log n)$ 。

Floyd

记原图中 u,v 之间边的边权为 $val\,(u,v)$ 。

我们注意到 Floyd 算法有一个性质：在最外层循环到点 k 时（尚未开始第 k 次循环），最短路数组 dis 中， $dis_{u,v}$ 表示的是从 u 到 v 且仅经过编号在 $[1,k)$ 区间中的点的最短路。

由最小环的定义可知其至少有三个顶点，设其中编号最大的顶点为 w，环上与 w 相邻两侧的两个点为 u, v，则在最外层循环枚举到 $k = w$ 时，该环的长度即为 $dis_{u,v} + val(v, w) + val(w, u)$。

故在循环时对于每个 k 枚举满足 $i < k, j < k$ 的 (i, j)，更新答案即可。

时间复杂度：$O(n^3)$

下面给出 C++ 的参考实现：

```cpp
int val[maxn + 1][maxn + 1];  // 原图的邻接矩阵
inline int floyd(const int &n) {
  static int dis[maxn + 1][maxn + 1];  // 最短路矩阵
  for (int i = 1; i <= n; ++i)
    for (int j = 1; j <= n; ++j) dis[i][j] = val[i][j];  // 初始化最短路矩阵
  int ans = inf;
  for (int k = 1; k <= n; ++k) {
    for (int i = 1; i < k; ++i)
      for (int j = 1; j < i; ++j)
        ans = std::min(ans, dis[i][j] + val[i][k] + val[k][j]);  // 更新答案
    for (int i = 1; i <= n; ++i)
      for (int j = 1; j <= n; ++j)
        dis[i][j] = std::min(
            dis[i][j], dis[i][k] + dis[k][j]);  // 正常的 floyd 更新最短路矩阵
  }
  return ans;
}
```

例题

GDOI2018 Day2 巡逻

给出一张 n 个点的无负权边无向图，要求执行 q 个操作，三种操作

1. 删除一个图中的点以及与它有关的边
2. 恢复一个被删除点以及与它有关的边
3. 询问点 x 所在的最小环大小

对于 50% 的数据，有 $n, q \leq 100$

对于每一个点 x 所在的简单环，都存在两条与 x 相邻的边，删去其中的任意一条，简单环将变为简单路径。

那么枚举所有与 x 相邻的边，每次删去其中一条，然后跑一次 Dijkstra。

或者直接对每次询问跑一遍 Floyd 求最小环， $O(qn^3)$

对于 100% 的数据，有 $n, q \leq 400$ 。

还是利用 Floyd 求最小环的算法。

若没有删除，删去询问点将简单环裂开成为一条简单路。

然而第二步的求解改用 Floyd 来得出。

那么答案就是要求出不经过询问点 x 的情况下任意两点之间的距离。

怎么在线?

强行离线，利用离线的方法来避免删除操作。

将询问按照时间顺序排列，对这些询问建立一个线段树。

每个点的出现时间覆盖所有除去询问该点的时刻外的所有询问，假设一个点被询问 x 次，则它的出现时间可以视为 $x+1$ 段区间，插入到线段树上。

完成之后遍历一遍整棵线段树，在经过一个点时存储一个 Floyd 数组的备份，然后加入被插入在这个区间上的所有点，在离开时利用备份数组退回去即可。

这个做法的时间复杂度为 $O(qn^2 \log q)$ 。

还有一个时间复杂度更优秀的在线做法。

对于一个对点 x 的询问，我们以 x 为起点跑一次最短路，然后把最短路树建出来，顺便处理出每个点是在 x 的哪棵子树内。

那么一定能找出一条非树边，满足这条非树边的两个端点在根的不同子树中，使得这条非树边 + 两个端点到根的路径就是最小环。

证明:

显然最小环包含至少两个端点在根的不同子树中一条非树边。

假设这条边为 (u,v)，那么最短路树上 x 到 u 的路径是所有 x 到 u 的路径中最短的那条， x 到 v 的路径也是最短的那条，那么 $x \to u \to v \to x$ 这个环肯定不会比最小环要长。

那么就可以枚举所有非树边，更新答案。

每次询问的复杂度为跑一次单源最短路的复杂度，为 $O(n^2)$ 。

总时间复杂度为 $O(qn^2)$ 。

评论