

实验 4.1：保护御坂网络

V1.1

实验背景

网络的某些节点遭到了黑客入侵，黑客在这些节点上放置了病毒程序。病毒的工作方式是这样的：病毒会自动感染与这个节点直接相连的所有节点。且病毒会持续扩散传播，直到没有更多的节点可以被这种方式感染。

现在为了降低网络中被感染的节点数量，我们可以关闭一个初始被入侵的节点（关闭某个节点之后，这个节点会断开所有连接，病毒无法入侵此节点，也不能穿过此节点感染其他节点）。请求出哪个节点可以满足要求。

不说人话的部分

我们定义点集在无向图中的连通分量为：点集中所有的点所在的连通分量的并集。

现有图 $G(V, E)$ ， $V' \subseteq V$ 。现求点 $v \in V'$ ，使得 $V' - v$ 在 $G - v$ 中的联通分量所含点数最少。

说人话的部分

1. 输入：若干行。
 - a. 第 1 行：网络节点数量 n ，即 $Card(V)$ 。本行以回车结束。
 - b. 第 2 ~ 第 $n+1$ 行：图的邻接矩阵。矩阵的每列用空格隔开，每行用回车隔开。
 - c. 第 $n+2$ 行：初次被感染的节点数量 m ，即 $Card(V')$ 。本行以回车结束。
 - d. 第 $n+3$ 行：初次被感染的节点，即 V' 。不同的节点用空格隔开。本行以回车结束。
2. 输出：
 - a. 只有一行，代表被删除的节点（即 v ）。输出完毕后不要加回车。
 - b. 如果删除任何一个节点后的最终感染数量与不删除节点的最终感染数量相同，请输出-1。
 - c. 如果多个节点都满足条件（即删除两个不同节点，最后的结果都一样），请输出编号最小的节点。
 - d. b 的优先级高于 c。
3. 提示与要求：
 - a. 节点编号从 0 开始；
 - b. 输入的图未必是连通图；
 - c. 设输入的邻接矩阵为 G 。我们保证：
 - 1) G 是对称矩阵。
 - 2) $G[i][j]$ 只会为 0 或 1，且 $G[i][i] = 1$ 。这么做的原因是本机与自己是直接相连的；
 - 3) 被感染的节点编号 $< n$ 。编号的输入是无序的；
 - 4) 不会一上来就感染所有节点，即保证 $m < n, V' \subsetneq V$ 。
 - d. 下周（14 号之后）我们还会放一道题，两道题的截止日期相同，所以请合理安排时间。
这两道题中，由于本题难度较大（准确地说，是比较绕），两道题的分值占比会进行动态分配。具体占比暂时不会公布，但是可以确定的是，4.1 ac 的人越多，占比越高。
 - e. 通过测点数量与最终得分的映射关系暂未确定，之后会发布。
我们按过的测试点数量给分，而不是按用时、尝试次数、board 排名等给分。但我们鼓励各位写出简洁、优美、高效的代码。
 - f. 本次实验不准使用 STL。
4. 数据规模说明：

50%的测点满足： $0 < m < n \leq 20$ 。

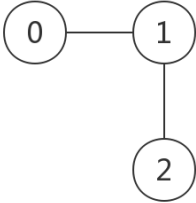
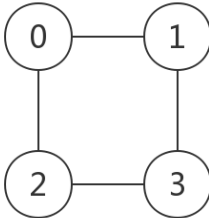
70%的测点满足： $0 < m < n \leq 500$ 。

100%的测点满足： $0 < m < n \leq 1500$ 。

注:

- 我们当然想把数据规模设置为 $0 < m < n \leq 9969$, 但是如果那样的话规模就太大了, oj 上会超时。
- 经过测试, 在 $n = 1500$ 的条件下, 助教的程序(预期解)用时在 200ms 左右 (没用 iostream 和 STL), 本用时仅供参考。如果您超时, 请考虑一下算法时间复杂度的问题。
- 如果你的方法过于暴力, 建议使用 `getchar` 读取邻接矩阵, 缩短读入时间。

输入输出样例

输入	输出	图的示意	解释
3 1 1 0 1 1 1 0 1 1 2 0 1	1		如果删掉 0, 1 的病毒还会继续扩散到 2; 如果删掉 1, 2 就不会被感染。
4 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 2 2 1	-1		无论删掉 1 还是 2, 节点 0 和 3 最后都会被感染, 因此输出 -1.

除非实验要求出了 bug 导致实验要求需要更新, 我们不会提供其他测试样例。

Deadline

考虑到各位 24 号有考试, DDL 改为 2019-11-27 23:59 (UTC+08:00)。只推一天的原因是周一和周二都能写, 而且 4.2 很简单。

除非后续由老师说明, 本次实验无实验报告。

提交方式

在 oj.ustc.edu.cn 上提交。

STAFF

既然反响不错, 那就继续搞事。祝大家能够愉快的度过本门课程。

原作、脚本、题目、测试: 毛浩宇

协力: 李鑫