

网络流与二分图匹配基础入门

cssyz-wjj

网络流

想象有一家自来水厂（源点， S ）可以供应无穷多的水，与你家（ T ，汇点）能吸收无穷多的水。 S 与 T 之间有一个 **有向图** 连接的管道网络，管道之间有结点连接。

单位时间 内，结点上能流过无穷多的水，但是对于管道 j ，它只能 **单向流过至多** c_j （容量）的水。另外，假设管道 j 实际流过的水为 f_j 。

现在我们需要对每个管道指定一个流量，满足以下条件即为这个网络的一个 **可行流**：

1. 每个结点流入的流量与流出的流量相同（除 S , T ）
2. $f_j \leq c_j$

如果单位时间流入 T 的流量最大，那我们称这个最大的流量为这个网络的 **最大流**。

为了解决这个问题，我们需要引入几个概念。

残量网络：对于一条边，我们令它的残量为 $g = c - f$ （有更严谨的表达方法，这里为了方便这么写），然后构成的图为

反图（好像是叫这么名字，而且具体概念我记不太清了）：就是对于单向边 (u, v, c) ，我们对应地建立一张 (v, u, c) 的单向边构成的图，这条边可以视作一条管道（如果带费用就会有别的需要注意的地方）

增广路：这是最重要的概念，但是这里口述

反悔：带上反图跑。这样找不到增广路那么就是最大流了，可以感性理解一下？

1. 建图, 对于边 $(u, v, g = c)$, 同时建立 $(v, u, 0)$, 注意这里的第三维实际就是残量 (省去了别的实际不需要的元素)
2. 在残量网络上找到一条增广路增广 (残量为 0 的边不能跑, 可以视作删除这条边)
3. 重复 2, 直到找不到了

FF 就是每次用 dfs 找到一条增广路增广即可

EK 就是每次用 bfs 找到一条增广路增广即可 (即算有几次只找到了流量很小的增广路, 也没有关系)

Dinic 就是使得每次只能增广一条路，变为可以增广多条路

1. 建图
2. 用 bfs 求出残量网络上每个点到 S 的最短路
3. 然后假装你一开始有一个带着无穷多的水的快递员，沿着最短路（树）跑，但是如果这条边的残量不太够，就要卸下尽可能少的水使得能够过去（具体操作看代码）
4. 如果还有水剩就继续跑
5. 重复 2 到 4，直到找不到增广路

Dinic 代码见附加文件

费用流

用 EK 跑

EK（费用流）代码见附加文件

还有其他什么预流推进自己去学吧，本人不会（）

二分图最大匹配

找交错路，不过一般可以用网络流跑

代码见附加文件

二分图最大权完美匹配

KM

本人不记得了.....不过要代码可以贴一个

虽然说感觉 KM 的题目很少