网络流

To_Carpe_Diem

河南省实验中学 & Henan Experimental High School

目录

1. 自我介绍
2. 网络流基础
3. Ford-Fulkson 方法
4. EK & Dinic
5. 例题与建模

自我介绍

我是来自河南省实验中学高一 (2) 班的史曜铭

洛谷 & uid：To_Carpe_Diem & 837396

博客园：To_Carpe_Diem

流网络和流

为了更加准确的描述网络流问题，我们先在这里引入一些 OI 竞赛中核心的记号、定义和理论引理。

流网络和流

为了更加准确的描述网络流问题，我们先在这里引入一些 OI 竞赛中核心的记号、定义和理论引理。

对于流网络 $G = (V, E)$ 是一个有向图，每条边 $(u, v) \in E$ 都有一个非负的容量值 $c(u, v) \geq 0$，而且如果边集 E 包含一条 (u, v)，则不存在反向边 (v, u)。

流网络和流

为了更加准确的描述网络流问题，我们先在这里引入一些 OI 竞赛中核心的记号、定义和理论引理。

对于流网络 $G = (V, E)$ 是一个有向图，每条边 $(u, v) \in E$ 都有一个非负的容量值 $c(u, v) \geq 0$，而且如果边集 E 包含一条 (u, v)，则不存在反向边 (v, u)。如果 $(u, v) \notin E$，方便起见，我们定义 $c(u, v) = 0$，并且在图中不允许自循环，在流网络中的所有节点中，我们特别的定义两个特殊节点：**源节点** s **和汇点** t。同时，为了方便，我们假定每个节点都在源点到汇点的某条路径上，即存在一条路径 $s \to v \to t$。

流网络和流

为了更加准确的描述网络流问题，我们先在这里引入一些 OI 竞赛中核心的记号、定义和理论引理。

对于流网络 $G = (V, E)$ 是一个有向图，每条边 $(u, v) \in E$ 都有一个非负的容量值 $c(u, v) \geq 0$，而且如果边集 E 包含一条 (u, v)，则不存在反向边 (v, u)。如果 $(u, v) \notin E$，方便起见，我们定义 $c(u, v) = 0$，并且在图中不允许自循环，在流网络中的所有节点中，我们特别的定义两个特殊节点：**源节点** s **和汇点** t。同时，为了方便，我们假定每个节点都在源点到汇点的某条路径上，即存在一条路径 $s \to v \to t$。因此来说，流网络是连通的，我们有 $|E| \geq |V| - 1$。

流网络和流

我们的网络 G 中的流实际上是一个实值函数 $f\colon V\times V\to R$，且其满足下面的两条性质：

流网络和流

我们的网络 G 中的流实际上是一个实值函数 $f\colon V\times V\to R$，且其满足下面的两条性质：

容量限制： 对于所有的节点 $u,v\in V$，要求 $0\le f(u,v)\le c(u,v)$。

流量守恒： 对于所有的节点 $u\in V-\{s,t\}$，要求 $\sum_{v\in V}f(v,u)=\sum_{v\in V}f(u,v)$。
特别的，若 $(u,v)\notin E$，那么 $f(u,v)=0$。

流网络和流

我们的网络 G 中的流实际上是一个实值函数 $f\colon V\times V\to R$，且其满足下面的两条性质：

容量限制： 对于所有的节点 $u,v\in V$，要求 $0\le f(u,v)\le c(u,v)$。

流量守恒： 对于所有的节点 $u\in V-\{s,t\}$，要求 $\sum_{v\in V}f(v,u)=\sum_{v\in V}f(u,v)$。
特别的，若 $(u,v)\notin E$，那么 $f(u,v)=0$。由于 $f(u,v)$ 表示节点 u 到 v 的一个非负流，那么我们的 G 的一个流 $|f|$ 可以做一下的定义：

$$|f|=\sum_{v\in V}f(s,v)-\sum_{v\in V}f(v,s)$$

也就是说，流出源点的量减去流出源点的量 11 即为这个流 f 的总流量。

反平行边

由于我们的定义，如果边 $(u, v) \in E$，那么 $(v, u) \notin E$，那倘若真出现了这种反边的情况，我们称之为反平行。因此我们必须把这种情况转换掉，我们可以考虑拆边，将这个边变为一个点，再用两条与原来相同的边分别连向这个点，和连出去。

反平行边

由于我们的定义，如果边 $(u, v) \in E$，那么 $(v, u) \notin E$，那倘若真出现了这种反边的情况，我们称之为反平行。因此我们必须把这种情况转换掉，我们可以考虑拆边，将这个边变为一个点，再用两条与原来相同的边分别连向这个点，和连出去。

我们容易得到这样的图的等价的。(若有需要请自证)

残留网络

从直观上来看，我们如果有一个网络 G 和流量 f，残留网络的 G_f 由那些仍有空间可以进行调整的边，我们将其残留容量记为 $c_f(u,v) = c(u,v) - f(u,v)$。若是 $c_f(u,v) = 0$，则该边不属于 G_f。

残留网络

从直观上来看，我们如果有一个网络 G 和流量 f，残留网络的 G_f 由那些仍有空间可以进行调整的边，我们将其残留容量记为 $c_f(u,v) = c(u,v) - f(u,v)$。若是 $c_f(u,v) = 0$，则该边不属于 G_f。对于流网络 $G = (V, E)$ 中的一个可行流 f，残量 $c_f(u,v)$ 的形式化定义为：

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{若}(u,v) \in E \\ f(v,u) & \text{若}(v,u) \in E \\ 0 & \text{其他} \end{cases}$$

所以我们的 G 的残留网络 $E_f = \{(u,v) \in V \times V : c_f > 0\}$，这是显然的，符合我们刚刚所说的定义。

引理

若 G_f 中存在一个流 f'，那么我们将这个流加入原流，形式化的讲，即 $|f \uparrow f'| = |f| + |f'|$，看起来很显然，可以自己尝试证明，不过过程挺麻烦的，参见《算法导论》P420。

增广路径

给定流网络 $G = (V, E)$ 和流 f，增广路径 p 即为残留网络 G_f 的一条从源点到汇点的简单路径。我们想要的是最大化路径上最小的 $c_f(u, v)$。我们称一条增广路径上的 p 的能增加的流量最大值为 $c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$

增广路径

给定流网络 $G = (V, E)$ 和流 f，增广路径 p 即为残留网络 G_f 的一条从源点到汇点的简单路径。我们想要的是最大化路径上最小的 $c_f(u, v)$。我们称一条增广路径上的 p 的能增加的流量最大值为 $c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$
接下来我们要证明一个推论，这是我们的 Ford-Fulkson 方法的复杂度核心：

$$|f \uparrow f_p| = |f| + |f_p| > |f|$$

这是显然的，就不证明了，带入刚刚的引理和 $c_f(p)$ 就能立刻得到这个结论。

流网络的割

流网络 $G = (V, E)$ 中的一个切割 (S, T) 将点集合 V 划分为 S 和 $T = V - S$ 的两个集合。若 f 是 G 的一个流，那么定义横跨切割的流量 $f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$

流网络的割

流网络 $G = (V, E)$ 中的一个切割 (S, T) 将点集合 V 划分为 S 和 $T = V - S$ 的两个集合。若 f 是 G 的一个流，那么定义横跨切割的流量 $f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$ 切割 (S, T) 的容量即为 $\sum_{u \in S} \sum_{v \in T} c(u, v)$。

最大流最小割定理

首先我们要证明的是，对于任意的一个割 (S, T)，则 $f(S, T) = |f|$。

最大流最小割定理

首先我们要证明的是，对于任意的一个割 (S, T)，则 $f(S, T) = |f|$。
这个我们直接展开相消即可得到。
设 f 为流网络 $G = (V, E)$ 的一个流，那么下面的内容为等价的：

f 为 G 的最大流

残留网络 G_f 不包括任何增广路径 p

$|f| = c(S, T)$，其中 (S, T) 是流网络 G 的某个切割。

Ford-Fulkson 基本算法

在我们的 Ford-Fulkson 的每次迭代中，我们每次找到一条增广路径 p，用 $f \uparrow f_p$ 来替换 f，从而得到一个 $|f| + |f_p|$ 的更大的流，直到找不到增广路。
我们需要在残余网络上对于边进行修改，对于正向的就减去，反向就加上这个容量。
我们可以来看一下精美的伪代码。

Ford-Fulkson 基本算法

FORD-FULKERSON(G, s, t)

```
for each edge (u, v) ∈ G.E do
    (u, v).f = 0
    while there exists a path p from s to t in the residual network G_f do
        c_f(p) = min{c_f(u, v) : (u, v) is in p}
        for each each edge (u, v) in p do
            if (u, v) ∈ E then
                (u, v).f = (u, v).f + c_f(p)
            else
                (v, u).f = (v, u).f − c_f(p)
```

不难发现复杂度是 $\mathcal{O}(E|f^*|)$ 的。

EK 算法

EK 算法，本质上的区别就是在增广的时候找最短路进行增广。
在 EK 算法中，由于我们每次都使用 BFS 寻找最短增广路（边权视为 1），源点 s 到任意节点 v 的最短距离 $d(s, v)$ 在每次增广后绝不会减少。
在一次增广过程中，如果边 (u, v) 是增广路上容量最小的边（即瓶颈），那么在增广后该边会从残量网络中消失。我们称这条边为该轮的关键边。

EK 算法

EK 算法，本质上的区别就是在增广的时候找最短路进行增广。
在 EK 算法中，由于我们每次都使用 BFS 寻找最短增广路（边权视为 1），源点 s 到任意节点 v 的最短距离 $d(s,v)$ 在每次增广后绝不会减少。
在一次增广过程中，如果边 (u,v) 是增广路上容量最小的边（即瓶颈），那么在增广后该边会从残量网络中消失。我们称这条边为该轮的关键边。
要让关键边 (u,v) 重新出现在残量网络中，必须先有一条增广路经过它的反向边 (v,u)。当 (u,v) 是关键边时，满足：
$d(s,v)=d(s,u)+1$。要让 (u,v) 重现，必须先增广 (v,u)，此时需满足：$d'(s,u)=d'(s,v)+1$。由于距离非递减，$d'(s,v)\geq d(s,v)$。代入得：$d'(s,u)\geq d(s,v)+1=d(s,u)+2$。这意味着，每当一条边重新成为关键边时，$s$ 到 u 的最短距离至少增加 2。

EK 算法

单条边的贡献：从 s 到 u 的最大距离约为 V。因为每次重现距离加 2，所以每条边 (u, v) 最多只能成为关键边 $\frac{V}{2}$ 次。

总增广次数：网络中一共有 $2E$ 条边（计入反向边），因此总增广次数最多为 $E \times \frac{V}{2} = O(VE)$。

单次增广开销：每次 BFS 寻找增广路的时间复杂度为 $O(E)$。

总复杂度：$O(VE) \times O(E) = \mathbf{O(VE^2)}$。

EK 算法

单条边的贡献：从 s 到 u 的最大距离约为 V。因为每次重现距离加 2，所以每条边 (u, v) 最多只能成为关键边 $\frac{V}{2}$ 次。

总增广次数：网络中一共有 $2E$ 条边（计入反向边），因此总增广次数最多为 $E \times \frac{V}{2} = O(VE)$。

单次增广开销：每次 BFS 寻找增广路的时间复杂度为 $O(E)$。

总复杂度：$O(VE) \times O(E) = \mathbf{O(VE^2)}$。

写板子环节。

Dinic 算法

Dinic 算法的本质就是多路增广，对残余网络分层，对于分层图多路增广。

一次 BFS 分层，一次 DFS 找完当前层所有增广路。

$$O(V) \text{ 轮 BFS} \times O(VE) \text{ 单轮 DFS} = \mathbf{O(V^2E)}$$

优化：当前弧优化、炸点

Dinic 算法

Dinic 算法的本质就是多路增广，对残余网络分层，对于分层图多路增广。

一次 BFS 分层，一次 DFS 找完当前层所有增广路。

$$O(V) \text{ 轮 BFS} \times O(VE) \text{ 单轮 DFS} = \mathbf{O(V^2E)}$$

优化：当前弧优化、炸点

写板子环节。

例题与习题

- P3376 【模板】网络最大流
- P2740 [USACO4.2] 草地排水 Drainage Ditches
- P2057 [SHOI2007] 善意的投票 / [JLOI2010] 冠军调查
- P2891 [USACO07OPEN] Dining G
- POJ1698 Alice's Chance
- U103816 【模板】最大权闭合子图
- P2762 太空飞行计划问题
- P1344 [USACO4.4] 追查坏牛奶 Pollutant Control
- P3153 [CQOI2009] 跳舞
- P3324 [SDOI2015] 星际战争
- P5038 [SCOI2012] 奇怪的游戏
- P3308 [SDOI2014] LIS
- POJ1149 PIGS
- [Non] 维护修改费 (Link 404)

讲解页面