

## 1 网络流建图

### 1.1 最小路径覆盖

**最小路径覆盖：**在一个有向无环图中，找出最少的路径，使得这些路径经过了所有的点。最小路径覆盖分为**最小不相交路径覆盖**和**最小可相交路径覆盖**，区别是这些路径是否可以相交

#### 1.1.1 最小不相交路径覆盖

**建图方法：**把原图的每个点  $u$  拆成两个点  $u_1, u_2$ ，如果有一条有向边  $(a, b)$ ，则连边  $(a_2, b_1)$ ，容易发现这是一个二分图，那么用下面的定理就可以求出答案

**定理：**最小路径覆盖 = 原图节点数 - 新图最大匹配数

**证明：**

一开始每个点都是一条路径，每次找一条匹配边，代表合并两条路径

由于路径不相交（即每个点的入度和出度至少有一个为 1），所以二分图上的边也不相交（如果相交则说明某个点的入度或出度大于 1），这正好是匹配的定义

每条匹配边代表答案  $-1$ ，所以最小路径覆盖 = 原图节点数 - 新图最大匹配数

#### 1.1.2 最小可相交路径覆盖

对原图传递闭包，即若原图中  $(u, v)$  连通，则增加边  $(u, v)$ 。这可以用 *Floyd* 算法  $O(n^3)$  实现。然后对新图做最小不相交路径覆盖即可。因为在原图中相交的路径在传递闭包后可以拆分成另一条边，这样就不相交了

#### 1.1.3 最多不相交路径

这种问题变化比较多，但都能表示成以下形式：已知一些路径，每个节点只能属于一条路径，求能选择多少条路径使它们不相交。

主要的方法是拆点，将一个点拆成两个，然后连边，容量表示该点最多经过次数

### 1.2 最小割

#### 1.2.1 最大权闭合子图

**定义：**有一个有向图，每一个点都有一个权值，选择一个权值和最大的子图，使得每个点的后继都在子图里面，这个子图就叫最大权闭合子图。

**建图方法：**从源点  $s$  向每个正权点连一条容量为权值的边，每个负权点向汇点  $t$  连一条容量为权值的绝对值的边，有向图原来的边容量全部为无限大。

**定理：**最大权闭合子图 = 所有正权点之和 - 最小割

**关键性质：**如果  $s$  与  $i$  有边，表示  $i$  在子图中。如果  $i$  与  $t$  有边，表示  $i$  不在于子图中。即：割掉  $s$  与  $i$  表示不选  $i$ ，割掉  $i$  与  $t$  表示选  $i$ 。

**性质 1：**原图之间的边一定不会被割掉

边权为无穷大，当然不会被选进最小割

**性质 2：**只有  $s$  到  $t$  不联通时，才得到最大权闭合子图

反证法：若  $s$  到  $t$  连通，则一定存在节点  $i, j$  使  $s$  到  $i$  有边， $i$  到  $j$  有边（引理 1）， $j$  到  $t$  有边。而根据性质 1： $i$  在子图中， $j$  不在子图中，这与最大权闭合子图的定义矛盾，证毕由引理 2 可得，图的一个割就是一个闭合子图

由于一个割的边权和 = 不选的正权点 + 选的负权点绝对值 = 不选的正权点 - 选的负权点  
一个割的边权和 = 不选的正权点 + 选的负权点绝对值 = 不选的正权点 - 选的负权点。

闭合子图 = 正权点 + 负权点 = 所有正权和 - 不选的正权点 + 选的负权点 = 所有正权和 - 割的边权和  
闭合子图 = 正权点 + 负权点 = 所有正权和 - 不选的正权点 + 选的负权点 = 所有正权和 - 割的边权和

显然割的边权和最小的时候得到最大权闭合子图，证毕

### 1.3 最小割

**定理：**二分图最大独立集 =  $n$  - 二分图最大匹配

其实二分图独立集是特殊的一种最大权闭合子图。我们根据上文“收益”的思想，把选某个点的收益看为 1，左部节点为正权点，右部节点为负权点。按照最大权闭合子图的方式建图，答案为 正权和 - 最小割 =  $n$  - 最小割 =  $n$  - 最大流。我们发现把最大权闭合子图中  $INF$  的边换成 1 也不影响答案，因为图中其他边的容量都为 1。这样图就变成了二分图匹配中的图，最大流 = 二分图最大匹配

### 1.4 最大密度子图

**定义：**图的密度是图上的边的数量除以点数。求密度最大的子图。

**建图：**看到平均数想到 01 分数规划。二分答案  $mid$ ，那么问题转化为判定是否存在

在一个子图，使得边数  $-mid \cdot$  点数  $>$

0

边数  $-mid \cdot$  点数  $> 0$ 。那么可以把每条边的权看成 1，每个点的权看成  $-mid$ ，限制是选择一条边就必须选择边连接的两个点。于是把边看成左部点，点看成右部点，跑最大权闭合子图，若答案  $> 0$ ，则合法。

## 1.5 二元关系最小割模型

定义：有若干个变量，每个变量有 2 种取值，有若干个限制，每个限制形如“若变量  $x = a, y = b$ ，就要付出  $c$  的代价”。最大化所有变量的值之和减去最小代价。

建图：每个变量建一个点， $S$  到  $x$  连边表示  $x$  的一种取值的代价， $x$  到  $T$  连边表示  $x$  的另一种取值的代价。对于一个限制，在两个点之间连边。边权需要列方程解出。

## 1.6 二分图带权匹配

定义：每条边有边权，求匹配边权值之和最大的匹配

建图：在边上加上权，跑费用流即可

## 1.7 最大权不相交路径

定义：每条路径有一个权值（一般是边权和），在 \*\* 不相交路径数最多的情况下 \*\*，最大化费用

建图：同最多不相交路径，在连接两个拆点的边上加上费用跑费用流即可

## 1.8 不等式差分模型（网络流解线性规划）

定义：对于一些不太好直接想到建图的问题，我们可以数学建模，列出方程然后用线性规划求解。这样的好处是思维量较小，只要做代数变换就可以建图，而不用考虑建图的实际意义。我们需要把式子做差，使得每个未知数仅在两个等式中出现。

根据网络流中每个点流量平衡的思想，我们可以把  $-x_i$  看成从点  $i$  流出  $x_i$  的流量， $+x_i$  看成流入  $x_i$  的流量。等式为 0 就代表流量平衡。

建图：每个等式为图中一个顶点，添加源点  $S$  和汇点  $T$ 。

如果一个等式右边为非负整数  $c$ ，从源点  $S$  向该等式对应的顶点连接一条容量为  $c$ ，权值为 0 的有向边；如果一个等式右边为负整数  $c$ ，从该等式对应的顶点向汇点  $T$  连接一条容量为  $c$ ，权值为 0 的有向边。

如果一个变量  $x_i$  在第  $j$  个等式中出现为  $x_i$ ，在第  $k$  个等式中出现为  $-x_i$ ，且在目标

函数里的系数为  $c_i$ ，从顶点  $j$  向顶点  $k$  连接一条容量为  $+\infty$ ，费用为  $c_i$  的有向边。

- 如果一个变量  $y_i$  在第  $j$  个等式中出现为  $y_i$ ，在第  $k$  个等式中出现为  $-y_i$ ，且在目标函数里没有出现，从顶点  $j$  向顶点  $k$  连接一条容量为  $+\infty$ ，权值为 0 的有向边。

## 1.9 有上下界的网络流

### 1.9.1 无源汇有上下界可行流

**定义：** 无源汇网络指的是没有源点和汇点，每个点都有入边和出边且满足流量守恒的网络。在这个网络上求一个流量方案，使得每条边的流量必须在  $[l_i, r_i]$  之间，且每个点流量守恒。

有上下界费用流的核心是”补偿”。我们先假设每条边的流量均为  $l_i$ ，那么一定会有一些点流量不守恒。现在我们需要构造一个附加网络，使得把附加网络 and 原网络叠加（即对应边流量相加）之后的图满足流量守恒。

因为 Dinic 只能求有源汇最大流，所以是不能直接求出附加网络的流量的。那么我们可以在附加网络上添加一些不在原网络上的边和点，来实现我们的限制。

记  $d_i = \text{点 } i \text{ 的入流} - \text{点 } i \text{ 的出流}$ ，然后建附加网络：

1. 新建源点  $ss$  和汇点  $tt$
2. 对于原图中的每条边  $e_i = (u, v)$ ，连边  $(u, v, r_i - l_i)$ ，也就是说附加网络包括原网络的边。
3. 新建边来满足流量守恒

若  $d_i = 0$  则该点流量平衡，不用处理

若  $d_i > 0$  则入流  $>$  出流，那么附加网络中  $i$  的出边需要增加流量，我们连边  $(ss, i, d_i)$ ，这样求最大流的时候出边的流量会增加  $d + i$ ，叠加后满足流量守恒

若  $d_i < 0$  则入流  $<$  出流，那么附加网络中  $i$  的入边需要增加流量，同理连边  $(i, tt, -d_i)$ ，这样求最大流的时候入边的流量会增加  $-d_i$ ，叠加后满足流量守恒

那么当且仅当步骤 3 中新建边满流时有解，总可行流为  $\max flow(ss, tt) + \sum l_i$ 。每条边在原图中流量 = 容量下界 + 附加流中它的流量

### 1.9.2 有源汇有上下界可行流

**定义：** 在有源汇网络上求一个流量方案，使得每条边的流量必须在  $[l_i, r_i]$  之间，且除源汇外每个点流量守恒。

设原网络的源和汇分别为  $s, t$  我们在原网络上加一条边  $(t, s, +\infty)$ ，相当于把到汇点的所有流量都流回源点，这样每个点流量都守恒。然后套无源汇的方法即可。注意总流量 =  $t$  到  $s$  的无穷边在原图中的流量

### 1.9.3 有源汇有上下界最大流和最小流

**定义：**在有源汇网络上求一个流量方案，使得每条边的流量必须在  $[l_i, r_i]$  之间，且除源汇外每个点流量守恒。在这个条件下使得总流量最大/最小。

先按上面的方法求出一个有源汇有上下界可行流。然后在附加网络上再跑一次  $s$  到  $t$  的最大流（注意不是  $ss, tt!$ ）。最大流 = 可行流 + 第二次跑的  $s$  到  $t$  最大流。

再跑一次最大流是因为附加网络上属于原图的边还有流量没被“榨干”。容易发现只要附加网络上不属于原图的边满流，属于原图的边怎么跑流量都是守恒的。因为第一次跑最大流已经保证所有点守恒，第二次跑最大流不会经过不属于原图的边，因此等价于对原图跑一次普通的最大流，除源汇外流量守恒。两次合起来总流量一定守恒，这就保证了正确性。

同理求最小流就跑一次  $t$  到  $s$  的最大流。最小流 = 可行流 - 第二次跑的  $t$  到  $s$  最大流。这是因为 Dinic 过程中反向边的流量增加等价于正向边的流量减少。

### 1.9.4 有源汇有上下界最小费用流

**定义：**在有源汇网络上求一个流量方案，使得每条边的流量必须在  $[l_i, r_i]$  之间，且除源汇外每个点流量守恒。每条边单位流量的费用为  $c_i$ 。在这个条件下使得总费用最小，费用定义同一般费用流。（不要求总流量最大）

这是有上下界费用流常被误解的一点，即最小费用流求的是 \*\* 费用最小的可行流 \*\*，而不是最大流。

因此按有源汇可行流的方法建图，把原图中的边带上费用。 $\text{ans} = \text{mincost}(ss, tt) + \sum l_i c_i$