# 网络流模型选讲

Hiram (wtd2)

2019 年 1 月 12 日

# 目录

# 说明

- 因为网络流算法本身是一种工具和方法，而不是算法竞赛考察的方向，所以本次内容将不涉及网络流求解算法。

# 说明

- 因为网络流算法本身是一种工具和方法，而不是算法竞赛考察的方向，所以本次内容将不涉及网络流求解算法。
- 如有不会网络流算法的同学，请自行在课后学习相关内容，包括但不限于：
  - 最大流求解算法和方案构造
  - 费用流求解方法和方案构造
  - 有上下界的网络流问题的建图方法

# 网络流图

- 如果带权有限的有向图 $G = (V, E)$ 满足如下条件，则称之为网络流图（或容量网络）：
- 有且仅有一个结点 $S \in V$ 入度为 0. 称为源点。
- 有且仅有一个结点 $T \in V$ 出度为 0. 称为汇点。
- $\forall (u, v) \in E \quad \exists c(u, v) \in R^+$，称为这条弧的容量。特别地，如果 $(u, v) \notin E$，可以假定 $c(u, v) = 0$.
- 通过容量网络中的一条弧 $(u, v)$ 的流量（或净流）记为 $f(u, v)$.

# 可行流

在容量网络中满足以下条件的网络流称为可行流:

- 容量限制 (Capacity Constraints):

$$f(u,v) \leq c(u,v)$$

- 流量守恒 (Flow Conservation):

$$\sum_{w \in V} f(u,w) = 0, \quad \text{when} \ \ u \neq S \ \ \text{and} \ \ u \neq T$$

# 最大流

一个流的流量 $|f|$ 定义为：

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

对于一个网络流图，它最大的可行流的流量即为它的最大流。

# 网络流的性质

在任意时刻，$G$ 的网络流都满足如下性质。

- 容量限制：
  通过一条弧的流量不能超过这条弧的容量上限。
  $\forall u, v \in V,\ f(u,v) \le c(u,v)$
- 反对称性：
  从一个结点 $s$ 到另一个结点 $t$ 的净流量一定是从 $t$ 到 $s$ 净流量的相反数。
  $\forall u, v \in V,\ f(u,v) = -f(v,u)$
- 流守恒：
  对于 $G$ 中任意一个结点 $u$，如果它既不是源点也不是汇点，那么它到相邻结点的所有流量之和为 $0$.
  $\forall u \in V - \{s,t\},\ \sum_{w \in V} f(u,w) = 0$

# 最大流

- 最大流的模型相对都比较直接，模型直观易懂。
- 有的题直接求最大流，有的题需要通过是否满流来判定。
- 根据”容量限制”和”流量守恒”的特点建模。

# 奶牛晚会

Source: Unknown

- $n$ 头奶牛要办一个新年晚会，每头牛可以提供若干种菜，一共有 $m$ 种不同的菜，编号 $1-m$。
- 第 $i$ 头牛有 $t_i$ 种菜，但只能从中选 $x_i$ 种提供，且每种菜**只能提供一份**。
- 第 $i$ 种菜最多需要 $k_i$ 份，也就是说如果第 $i$ 种菜超过 $k_i$ 份超过部分将会被扔掉。
- 求主办方最多能收到多少**份**菜。
- $n \le 200, m \le 100, \forall t_i, x_i, k_i \le 10$

# 奶牛晚会

Source: Unknown

- 每头奶牛建点，每种菜建点。
- s 向奶牛连容量为 $x_i$，奶牛向能提供的菜连容量为 1，菜向 t 连容量为 $k_i$。

# 圆桌问题

Source: 网络流 24 题 by byvoid

- 来自 $n$ 个国家的代表开会，每个国家代表数为 $c_i$，会场有 $m$ 张圆桌，每张桌子可容纳 $k_i$ 人。
- 同一个国家的代表不能在一张桌子上就餐，设计一个合法方案。
- $n, m \leq 500$

# 圆桌问题

Source: 网络流 24 题 by byvoid

- 按国家和圆桌建点。
- s 向国家连容量为 $c_i$，国家向每张圆桌连容量为 1，圆桌向 t 连容量为 $k_i$。
- 判断是否满流，DFS 得到流的方案。

# Magic Potion

Source: 2018-2019 ACM-ICPC Asia Regional Nanjing

- 有 $n$ 个英雄和 $m$ 个怪兽，每个英雄可以杀死的怪兽的编号是集合 $M_i$。
- 共有 $k$ 瓶特殊的药物，每个英雄最多服用一瓶药。
- 每个英雄默认可以杀死其集合中的一个怪兽，嗑药的英雄可以杀死两个怪兽。
- 求最多能杀死多少个怪兽。
- $1 \leq n, m, k \leq 500$

# Magic Potion

Source: 2018-2019 ACM-ICPC Asia Regional Nanjing

- 按英雄和怪兽建点，并建立补充源 s’。
- s 向每个英雄连容量 1，每个怪兽向 t 连容量 1，每个英雄向可杀死的怪兽连容量 1。
- s 向 s’ 连容量 $k$，s’ 向每个英雄连容量 1。
- 判断最大流是否等于 $m$。

# 奇怪的游戏

Source: SCOI 2012

- 给出一个 $n \times m$ 的棋盘，每个格子上有一个数 $v_{ij}$。
- 可以给相邻的两个数同时 $+1$，相邻指有公共边。
- 求最少多少次操作可以将棋盘上的数变为同一个数。
- 如果不可能，则输出 $-1$。
- $n, m \leq 40$

# 奇怪的游戏

Source: SCOI 2012

- 给棋盘黑白染色，设黑格个数为 $\mathrm{num}_0$，数的总和为 $\mathrm{sum}_0$，白格为 $\mathrm{num}_1, \mathrm{sum}_1$。
- 设最后所有数字都变成 $c$，则 $(\mathrm{num}_0) * c - \mathrm{sum}_0 = (\mathrm{num}_1) * c - \mathrm{sum}_1$。

# 奇怪的游戏

Source: SCOI 2012

- 给棋盘黑白染色，设黑格个数为 $\mathrm{num}_0$，数的总和为 $\mathrm{sum}_0$，白格为 $\mathrm{num}_1, \mathrm{sum}_1$。
- 设最后所有数字都变成 $c$，则 $(\mathrm{num}_0) * c - \mathrm{sum}_0 = (\mathrm{num}_1) * c - \mathrm{sum}_1$。
- 若 $\mathrm{num}_0 \neq \mathrm{num}_1$，则唯一可行的 c 是 $c = \frac{\mathrm{sum}_0 - \mathrm{sum}_1}{\mathrm{num}_0 - \mathrm{num}_1}$。
- 否则，若 $\exists c_0$ 满足条件，$\forall c \geq c_0$ 亦满足条件，即符合二分的性质。

# 奇怪的游戏

Source: SCOI 2012

- 考虑如何建图判定某个 $c$ 是否符合条件。
- s 向所有黑点连容量 $c - v_{ij}$，所有白点向 t 连容量 $c - v_{ij}$。
- 所有黑点向相邻白点连容量 $\infty$。
- 判断是否能够满流。

# 割的定义

- 一个 s-t 割 $C=(S,T)$ 是一种 $V$ 的划分使得 $s\in S$，$t\in T$。
- $C$ 的割集是集合 $\{(u,v)\in E\ :\ u\in S, v\in T\}$
- 一个 s-t 割的容量是

$$c(S,T)=\sum_{(u,v)\in(S\times T)\cap E} c_{uv}=\sum_{(i,j)\in E} c_{ij}d_{ij}$$

- 其中 $d_{ij}=\begin{cases}1 & i\in S \text{ 且 } j\in T\\ 0 & i\notin S \text{ 或 } j\notin T\end{cases}.$

# 最大流最小割定理

- 最小 s-t 割问题： 计算 $c(S,T)$ 的最小值。即找到 $S$ 和 $T$ 使 $s-t$ 割的容量达到它的最小值。
- 最大流最小割定理：
  对于一个网络流，从源点到目标点的最大的流量等于最小割的每一条边的和。
  即对于一个如果移除其中任何一边就会断开源点和目标点的边的集合的边的容量的总和。
- 最大流最小割定理是线性规划中的对偶问题的一种特殊情况。

# 最大流最小割定理

- 根据最大流最小割定理，我们可以用求解最大流的方法得到最小割。
- 相比最大流，最小割的建模相对更隐蔽，需要进行更多的转换。

# Order

Source: CEOI 2008

- 有 $n$ 个工作，$m$ 个机器，每种机器有一个价格，每个工作需要使用一些机器，完成后可以获得收入。
- 你可以选择购买一些机器，求最大收益。
- 收益 = 收入-支出。
- $n, m \leq 500$

# Order

Source: CEOI 2008

- 有 $n$ 个工作，$m$ 个机器，每种机器有一个价格，每个工作需要使用一些机器，完成后可以获得收入。
- 你可以选择购买一些机器，求最大收益。
- 收益 = 收入-支出。
- $n, m \leq 500$
- 如果每种机器还可以按次数租用，求最大收益。

# Order

Source: CEOI 2008

- 最大权闭合子图的裸题。

# Order

Source: CEOI 2008

- 最大权闭合子图的裸题。
- 一种合法的方案满足：不存在一项完成的工作有对应的机器没有购买。
- 考虑初始所有工作都完成且所有机器都不购买，最小化方案变合法的代价。
- s 向每项工作连收入的容量，每项工作向对应的机器连 $\infty$ 容量，机器向 t 连机器价格容量。
- 此时任意割满足条件，最小割即为最小的方案变合法的代价。

# Order

Source: CEOI 2008

- 最大权闭合子图的裸题。
- 一种合法的方案满足：不存在一项完成的工作有对应的机器没有购买。
- 考虑初始所有工作都完成且所有机器都不购买，最小化方案变合法的代价。
- s 向每项工作连收入的容量，每项工作向对应的机器连 $\infty$ 容量，机器向 t 连机器价格容量。
- 此时任意割满足条件，最小割即为最小的方案变合法的代价。
- 可租用时，只需要将 $\infty$ 改为对应代价。

# 汽车展览会

Source: Unknown

- $n \times m$ 的矩形停车场，要举办一个展览会。
- $(i, j)$ 可以停 $a_{i,j}$ 辆车，边相邻的格子不能同时放车。
- 求最多能停多少车。
- $n, m \leq 50$

# 汽车展览会

Source: Unknown

- 类似第一题的做法，先假设所有格子都可取。
- 给网格图黑白染色，则相邻的黑白格子不能同时取得。
- s 向黑色点连 $a_{ij}$ 容量，白色点向 t 连 $a_{ij}$ 容量，黑色点向相邻白色点连 $\infty$ 容量。
- 容易发现一个割即满足条件，最小割即为最小的合法化的代价。

# 风扇

Source: 某省选集训

- 有一台有 $n$ 个均匀分布的扇叶的风扇，其中几片扇叶掉下来了。
- 给定掉了的扇叶的位置，在保证所有扇叶的重心还在重心上时，最多能保留多少片扇叶。
- $n \leq 5000$ 且 $n$ 有不超过两个质因子。

# 风扇

Source: 某省选集训

- 若 $n$ 是一个质数，则显然只有全部保留和全部删掉两种可能。

# 风扇

Source: 某省选集训

- 若 $n$ 是一个质数，则显然只有全部保留和全部删掉两种可能。
- 考虑 $n = p^{\alpha}$，则 $\{i, i + p^{\alpha-1}, \cdots, i + (p-1) \cdot p^{\alpha-1}\}$ $(i \le p^{\alpha-1})$ 只有全部保留和全部删掉两种可能。

# 风扇

Source: 某省选集训

- 若 $n$ 是一个质数，则显然只有全部保留和全部删掉两种可能。
- 考虑 $n = p^{\alpha}$，则 $\{i, i + p^{\alpha-1}, \cdots, i + (p-1) \cdot p^{\alpha-1}\}$ $(i \le p^{\alpha-1})$ 只有全部保留和全部删掉两种可能。
- 进一步考虑 $n = p^{\alpha} \cdot q^{\beta}$ 的情况，我们可以将其按照两种方式分组：

$$\{i, i + \frac{n}{p}, \cdots, i + (p-1) \cdot \frac{n}{p}\} \ (i \le \frac{n}{p})$$

$$\{i, i + \frac{n}{q}, \cdots, i + (q-1) \cdot \frac{n}{q}\} \ (i \le \frac{n}{q})$$

# 风扇

Source: 某省选集训

- 显然，若 $\forall x$ 已经被删除，它所对应的两组扇叶中至少有一组需要被全部删除。
- 同时，不难发现对 $\forall x$ 都满足有其对应的一组全部被删除，则剩余扇叶满足条件。

# 风扇

Source: 某省选集训

- 显然，若 $\forall x$ 已经被删除，它所对应的两组扇叶中至少有一组需要被全部删除。
- 同时，不难发现对 $\forall x$ 都满足有其对应的一组全部被删除，则剩余扇叶满足条件。
- 考虑将这 $\frac{n}{p}+\frac{n}{q}$ 组数建点。
- 不妨 s 向 $\frac{n}{p}$ 个点连容量为该组剩余扇叶数的边，同理 $\frac{n}{q}$ 个点向 t 连边。
- 对于每个掉了的扇叶，其对应的两组至少有一组需要全部被删除，只需将其对应的两组连容量 $\infty$ 的边。
- 其最小割即为最少删除的节点。

# 费用流

- 费用流通常指最小费用最大流。在给网络的每条边赋一个费用，即通过这条边的每单位流量产生的费用的前提下，求出一个流量最大且费用最小的流。
- 费用流问题的建模和最大流问题十分类似，只不过新增了单位边的费用，使得问题的某些特性更容易表现出来（如单位费用）。

# 订货

Source: HAOI 2010

- 在第 $i$ 月该产品的订货单价为 $d_i$，需求量为 $a_i$，上个月月底未销完的单位产品要付存贮费用 $m$。
- 每月月初订购，订购后产品立即到货，进库并供应市场，于当月被售掉则不必付存贮费。
- 如果第 1 月月初和第 $n$ 月月底的库存量为零，仓库的最大容量为 $S$（跨月存储才需要使用仓库），问如何安排这 $n$ 个月订购计划，才能使成本最低？
- $n \leq 50$

# 订货

Source: HAOI 2010

- 经典的费用流模型。

# 订货

Source: HAOI 2010

- 经典的费用流模型。
- s 向每个月连 $(\infty, d_i)$ 的边，每个月向 t 连 $(a_i, 0)$ 的边。
- 第 $i$ 个月向第 $i+1$ 个月连 $(S, m)$ 的边，其费用流即为所求。

# 晨跑

Source: SDOI 2009

- 现在给出一张地图，地图中包含 $n$ 个十字路口和 $m$ 条街道，Elaxia 只能从一个十字路口跑向另外一个十字路口，街道之间只在十字路口处相交。
- Elaxia 每天从寝室出发跑到学校，保证寝室编号为 1，学校编号为 $n$。
- Elaxia 的晨跑计划是按周期（包含若干天）进行的，由于他不喜欢走重复的路线，所以在一个周期内，每天的晨跑路线都不会相交（在十字路口处），寝室和学校不算十字路口。
- 他希望在训练周期包含的天数最大的前提下，跑的总路程尽量短。
- $n \leq 200, m \leq 20000$

# 晨跑

Source: SDOI 2009

- 本题的核心是如何保证一个十字路口只经过一次。

# 晨跑

Source: SDOI 2009

- 本题的核心是如何保证一个十字路口只经过一次。
- 考虑通过拆点限流来解决问题，将每个十字路口拆成 $a_i, b_i$。$a_i$ 视为从 $i$ 号点出发，$b_i$ 视为到达 $i$ 号点。
- s 向 $a_1$ 连 $(\infty, 0)$，$b_n$ 向 t 连 $(\infty, 0)$，$b_i$ 向 $a_i$ 连 $(1, 0)$。
- $\forall (u, v) \in E$，$a_u$ 向 $b_v$ 连 $(1, \text{距离})$。
- 其费用流即为所求。

Thanks for your listening.