中国计算机学会
China Computer Federation

网 络 流 算 法 及 其 应 用

金靖

2021.02.02

最大流算法

Ford-Fulkerson方法

Edmond-Karp算法

Dinic算法

最大流-最小割定理

最大流算法的应用

二分图的匹配问题

上下界网络流

费用流算法

基于SPFA的实现

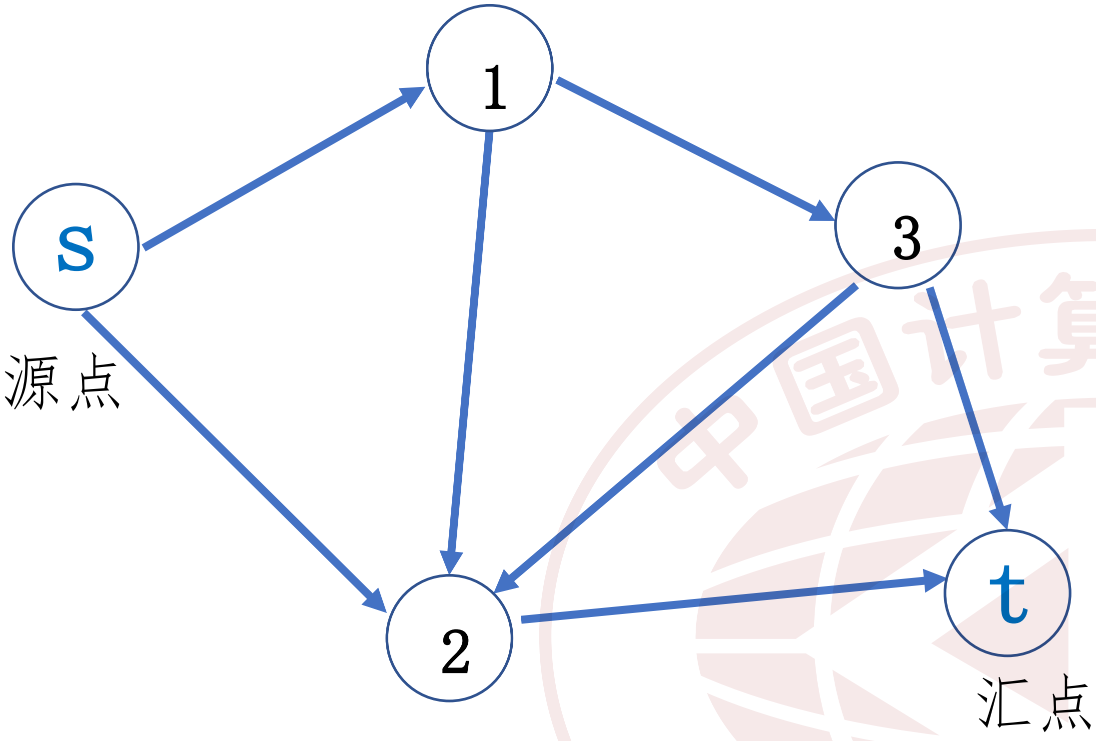
典型例题讲解

最小割模型的应用

典型例题讲解

最大权闭合图

二分图的最小点权覆盖集与最大点权独立集



$$G=(V,E)$$

$$e\in E,\begin{cases} c(e) & \text{容量} \\ f(e) & \text{流量} \end{cases}$$

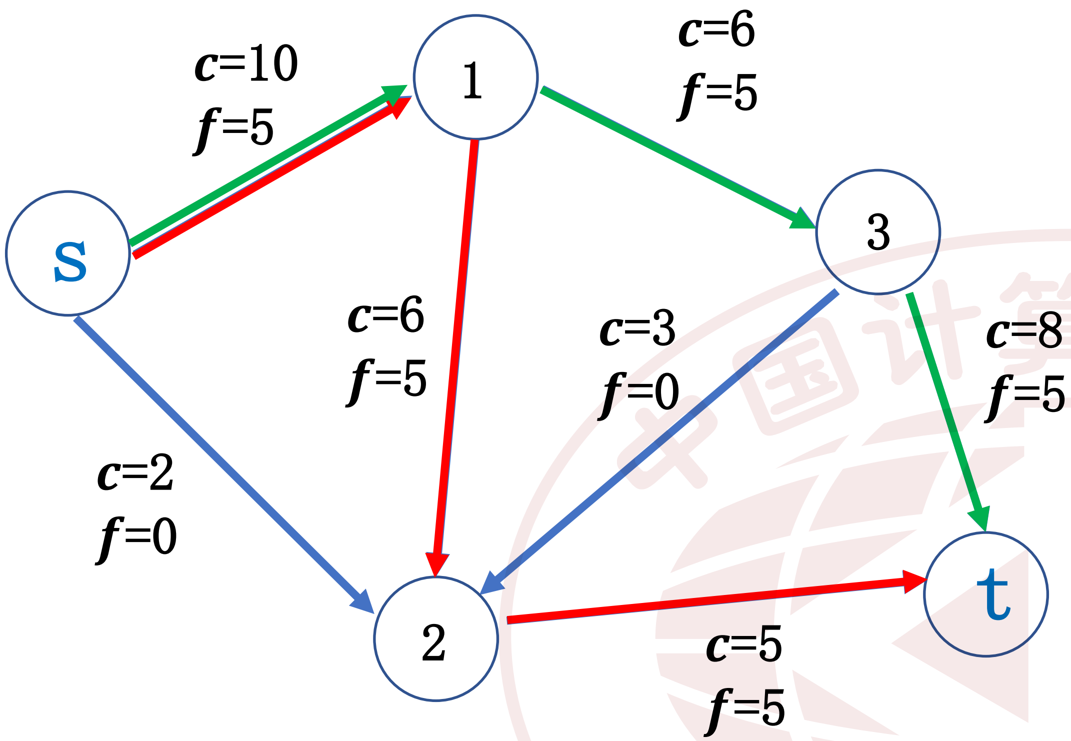
$$0\le f(e)\le c(e)$$ 容量限制

流网络 G

$\{$可行流$f\}$

最大可行流 $|f|_{max}$

$$V_x\in V-\{s,t\},\ \sum_{(v,x)\in E} f(v,x)=\sum_{(x,v)\in E} f(x,v)$$ 流量守恒



$\Rightarrow$ 最大流为10

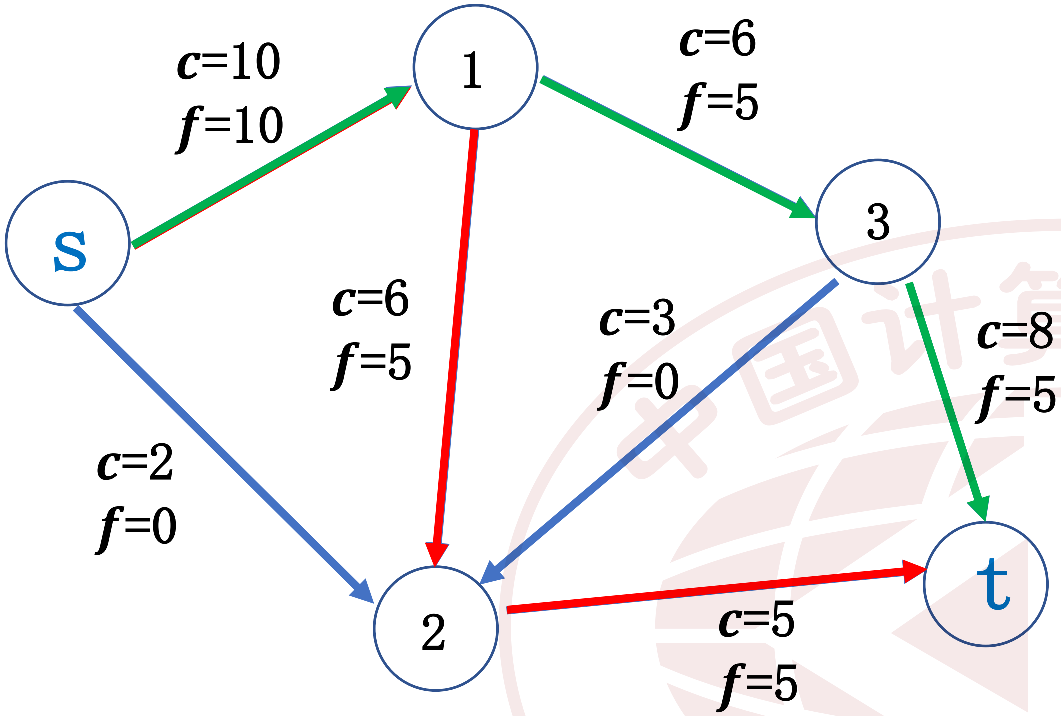
贪心算法：

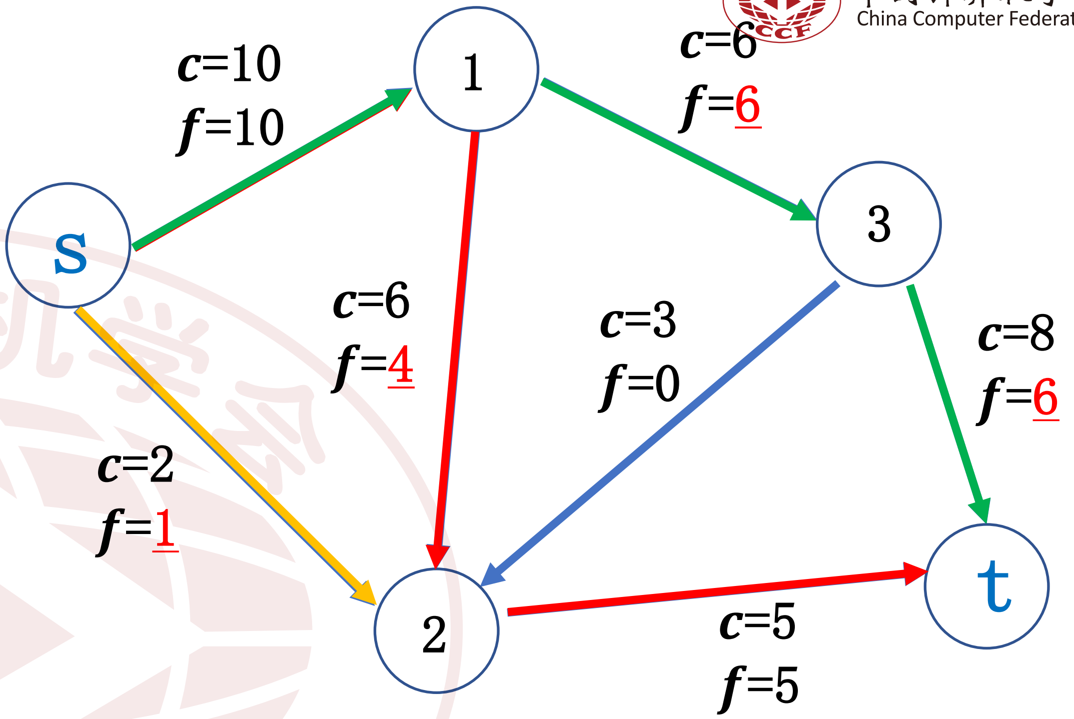
1、从 $s \to t$ 寻找一条路径，路径上所有的边满足容量限制 $f(e) \le c(e)$

2、如果不存在满足条件的路径，则结束算法。否则沿着这条路径尽可能的增加流量，返回第1步。

两个可行流方案相加，是一个新的可行流方案

$$|f_1 + f_2| = |f_1| + |f_2|$$





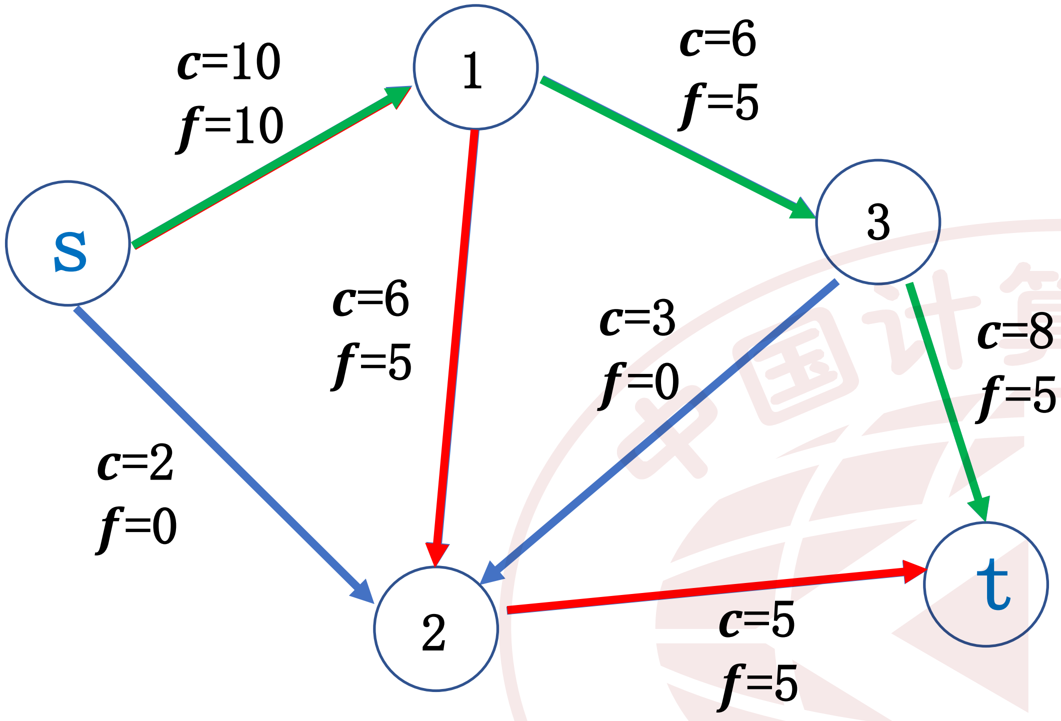
贪心算法：

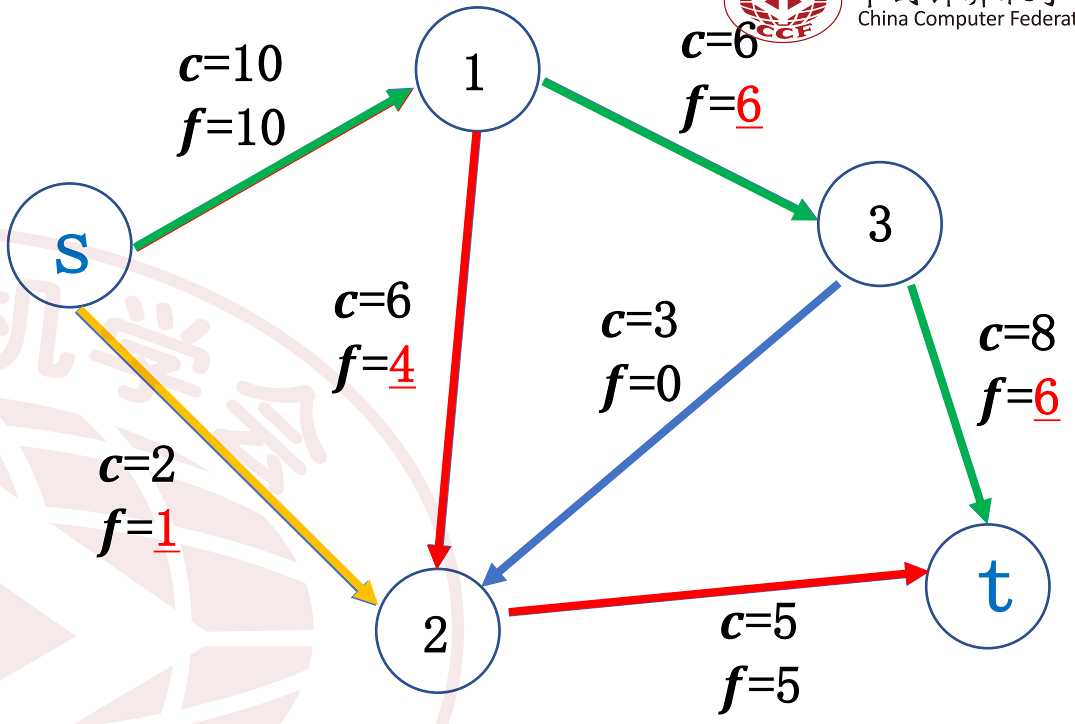
1、从 $s \rightarrow t$ 寻找一条路径，路径上所有的边满足容量限制 $f(e) \leq c(e)$

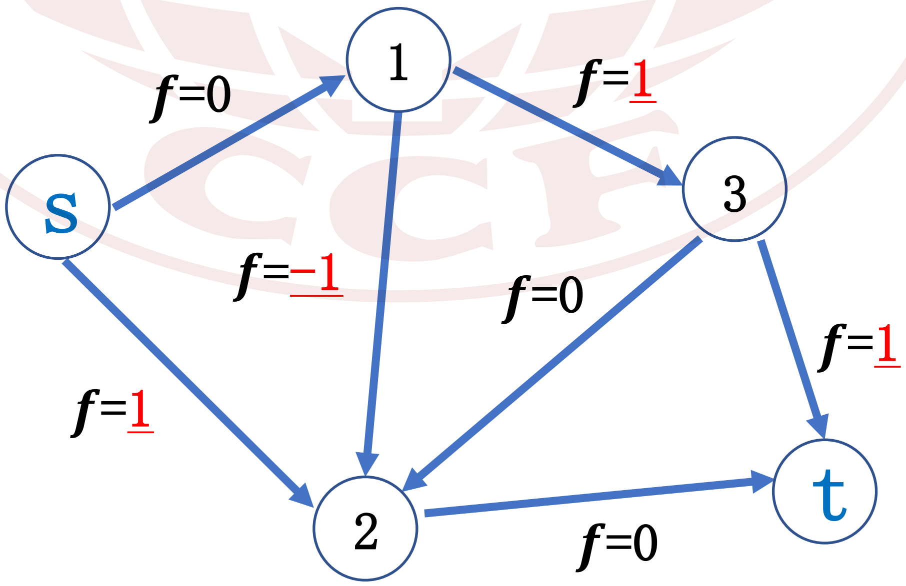
2、如果不存在满足条件的路径，则结束算法。否则沿着这条路径尽可能的增加流量，返回第1步。

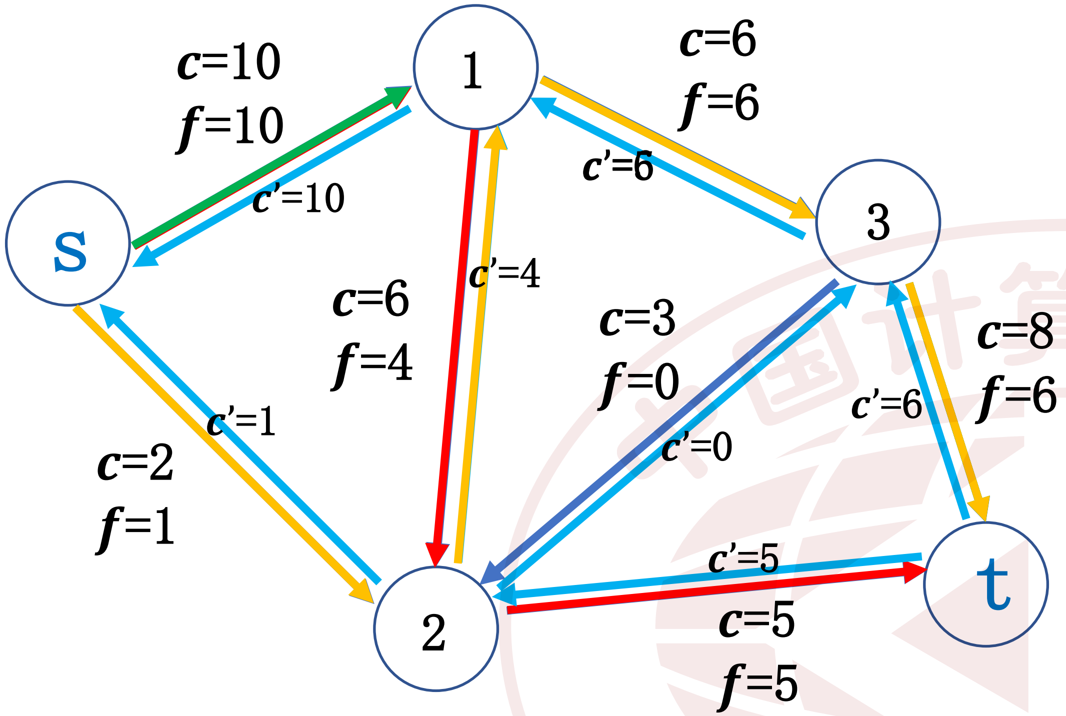
两个可行流方案相加，是一个新的可行流方案

$$|f_1 + f_2| = |f_1| + |f_2|$$


s
1
2
3
t
c=10
f=10
c=6
f=5
c=6
f=5
c=3
f=0
c=8
f=5
c=2
f=0
c=5
f=5


s
1
2
3
t
c=10
f=10
c=6
f=6
c=6
f=4
c=3
f=0
c=8
f=6
c=2
f=1
c=5
f=5


s
1
2
3
t
f=0
f=1
f=−1
f=0
f=1
f=1
f=0



在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图

残余网络 G_f

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

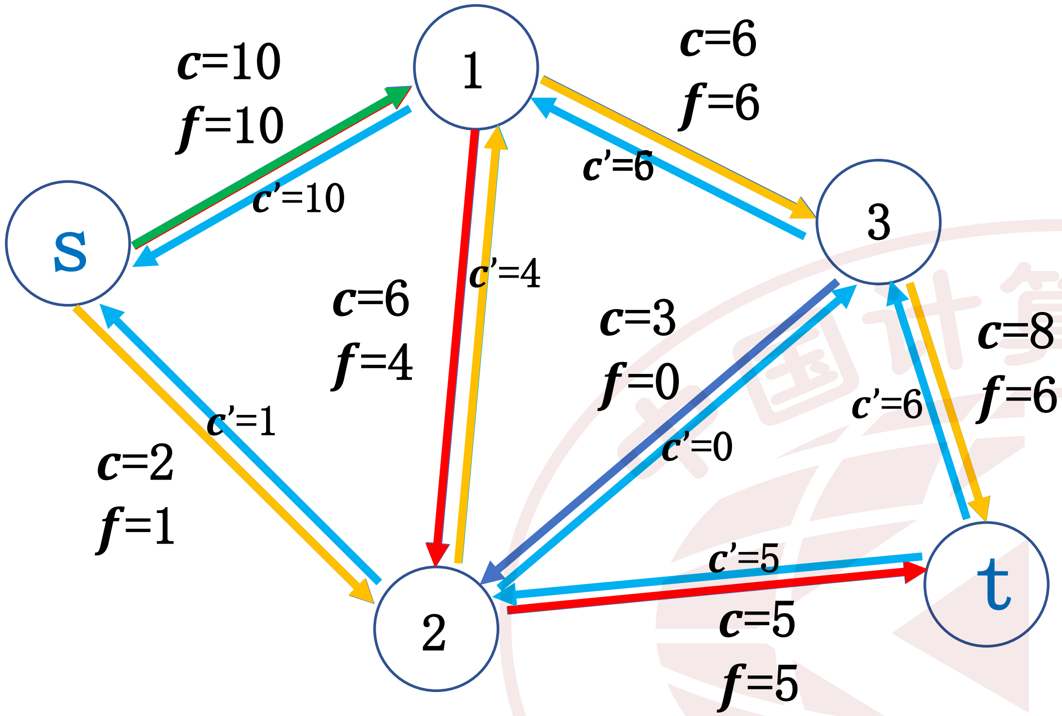
在残余网络上寻找 $s \rightarrow t$的路径

增广路

基于增广路的最大流方法

Ford-Fulkerson方法

残余网络中不存在增广路的充要条件是当前流已是最大流



在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图

残余网络 G_f

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

DFS寻找增广路

时间复杂度分析：

每次DFS找增广路为 $O(m)$；设最大流的流量为f，由于每次流量至少增加 1，总时间复杂度为 $O(fm)$ 。

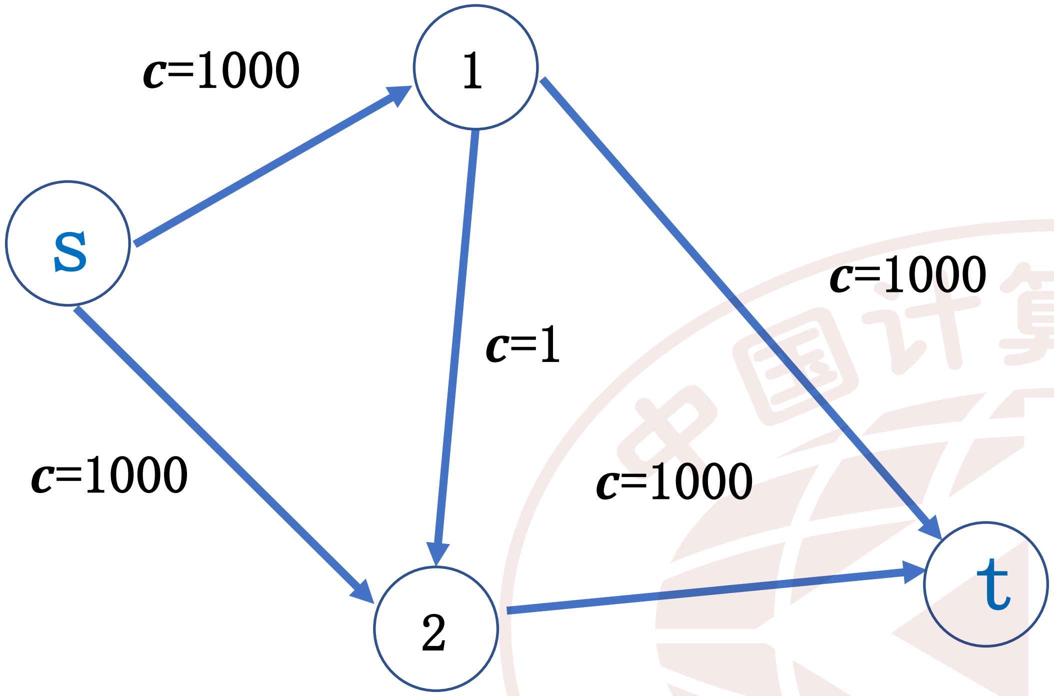
在残余网络上寻找 $s\to t$的路径

增广路

基于增广路的最大流方法

Ford-Fulkerson方法

残余网络中不存在增广路的充要条件是当前流已是最大流



在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图

残余网络 G_f

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

在残余网络上寻找 $s \to t$的路径

增广路

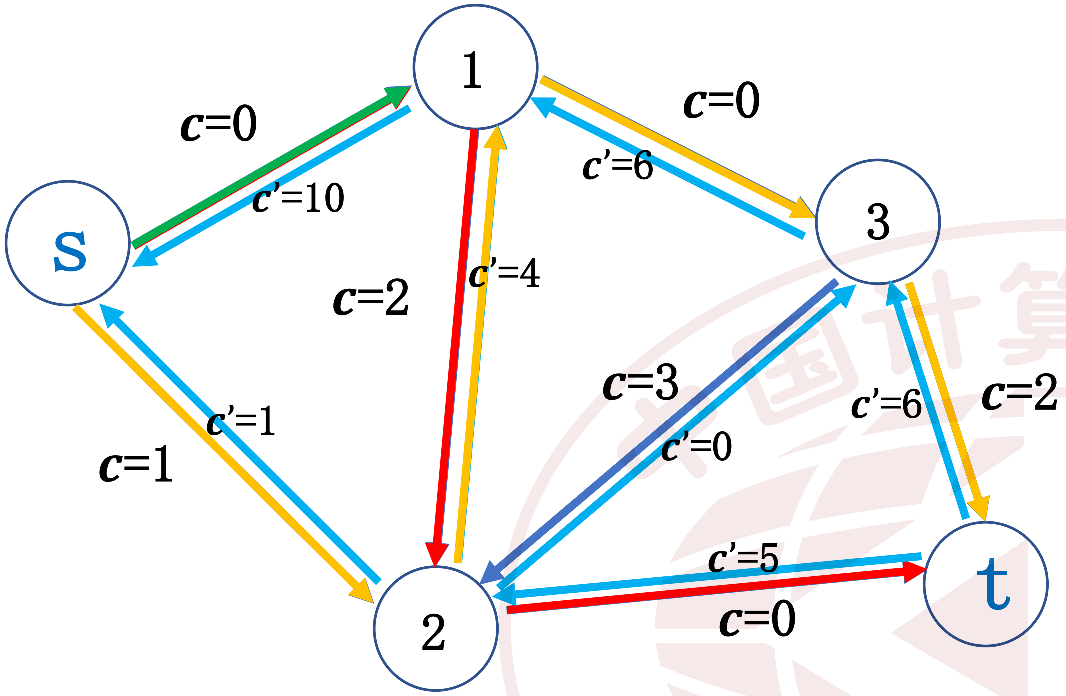
DFS寻找增广路

时间复杂度分析：
每次DFS找增广路为 $O(m)$；设最大流的流量为f，由于每次流量至少增加 1，总时间复杂度为 $O(fm)$ 。

基于增广路的最大流方法

Ford-Fulkerson方法

残余网络中不存在增广路的充要条件是当前流已是最大流



在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图

残余网络 G_f 记录每条边的剩余容量

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

BFS寻找增广路

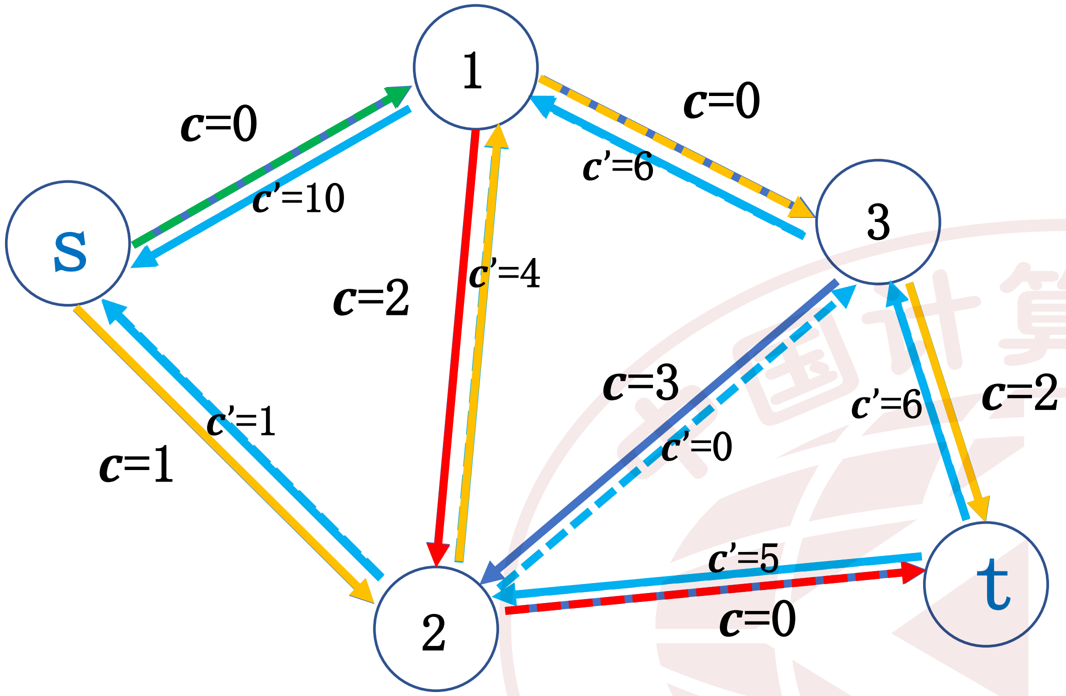
在残余网络上寻找 $s \to t$的路径

增广路

最短路增广算法

Edmonds-Karp算法

残余网络中不存在增广路的充要条件是当前流已是最大流



在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图

残余网络 $\boldsymbol{G_f}$ 记录每条边的剩余容量

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

BFS寻找增广路

在残余网络上寻找 $s \rightarrow t$的路径

增广路

最短路增广算法

Edmonds-Karp算法

残余网络中不存在增广路的充要条件是当前流已是最大流

时间复杂度分析：
每次对分层图进行多路增广的时间复杂度为 $O(nm)$。
而每做一次分层图后，最短增广路的长度会至少增加1，
由于增广路的长度不会超过$n-1$，因此最多重复$O(n)$次
分层图处理。
总时间复杂度为 $O(n^2m)$

基于分层图的多路增广算法**Dinic算法**

算法思想：

1. 在残余网络上，用BFS从源点s 到汇点t 构造分层图；
2. 在当前分层图上，使用DFS进行多路增广，在回溯时实时更新剩余容量。

直至在残余网络中，无法从源点s 到达汇点t。

在原图 G 中增添 **反向边**

根据原图的可行流 f，对应一个新图
残余网络 G_f 记录每条边的剩余容量

1. 可行流各不相同，残余网络也各不相同。
2. 反向边只存在于残余网络，在原图上并不存在。

在残余网络上寻找 $s \to t$ 的路径
增广路

残余网络中不存在增广路的充要条件是当前流已是最大流

最大流的算法

算法名称	复杂度	概要
增广路方法（Ford Fulkerson method）		
一般增广路算法	$O(nmU)$	在残余网络中，每次任意找一条增广路径增广。
容量缩放增广路算法	$O(nm\log U)$	在残余网络中，每次找一条有最大可增广容量和的增广路径增广，即残余网络中源到汇的最长路。
最短路增广算法（Edmond-Karp alogorithm）	$O(nm^2)$	在残余网络中，每次找一条含顶点数最少的增广路径增广，即残余网络中源到汇的BFS路径。
连续最短路增广路算法（Dinic algorithm）	$O(n^2m)$	在每次BFS找增广路时，记录每个点的距离标号（构造分层图）。在距离标号点所构成的最短路上，不断地通过DFS找增广路。即一次标号多次增广，以提高速度。
预流推进方法		
一般预流推进算法	$O(n^2m)$	维护一个预流，不断地对活跃节点执行push操作或relabel操作来调整这个预流，直到不能操作。
先进先出预流推进算法	$O(n^3)$	以先进先出队列维护活跃节点。
最高标号预流推进算法	$O(n^2\sqrt{m})$	每次检查具有最高标号的活跃节点。

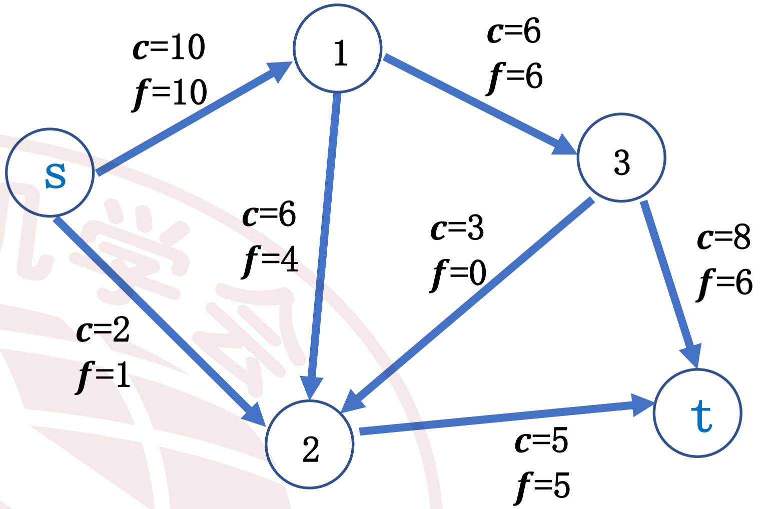
切割(cut)

- $G=(V,E)$的割：$[S,V-S]$
- 将点集 V 划分为 S 和 $V-S$ 两个部分
- $[S,V-S]$ 代表一个边集合
- 割的容量：边集中所有边的容量之和 $c(S,V-S)$
- 割的净流量(net flow)：边集中所有边的流量之和定义为 $f(S,V-S)$

如果 $s\in S, t\in(V-S)$，则称为$s-t$割$[S,T]$

对于一个网络，为了保证没有从 s 到 t 的路径，需要删去哪些边？

- $s-t$割的容量： $c(S,T)$
- $s-t$割的净流量： $f(S,T)$



S	T	$[S,T]$	$c[S,T]$	$f[S,T]$
$\{s,1,3\}$	$\{2,t\}$	$\{(s,2),(1,2),(3,2),(3,t)\}$	$2+6+3+8=19$	$1+4+0+6=11$
$\{s,1,2\}$	$\{3,t\}$	$\{(1,3),(2,t)\}$	$6+5-0=11$	$=$ $6+5=11$
$\{s,1,2,3\}$	$\{t\}$	$\{(2,t),(3,t)\}$	$8+5=13$	$6+5=11$
$\{s,2,3\}$	$\{1,t\}$	$\{(s,1),(2,t),(3,t)\}$	$10+5+8=23$	$10-4-6+6+5=11$

一个流网络 G 的可行流 f 的值，为从源点 s 向外发出的流量总和。

$$|f| = \sum_{v \in V} f[s,v]$$

流网络 G 中，设其任意一个流为 f，且 $[S,T]$ 为 G 的一个割，则通过割的净流量为 $f(S,T) = |f|$

$$f[S,T] = f(S,V) - f(S,S) = f(S,)$$
$$= f(s,V) + f(S-\{s\},V) = f(s,V) = |f|$$

通过割的净流就是网络的可行流

流网络 G 中，设任意一个流为 f，任意一个割为 $[S,T]$

$\because S$ 中除了源点以外的其他点，均流量守恒。

$\therefore |f|$为 S 的出边的总流量减去 S 的入边的总流量。

$$V_x \in V - \{s,t\}, \sum_{(v,x)\in E} f(v,x) = \sum_{(x,v)\in E} f(x,v)$$

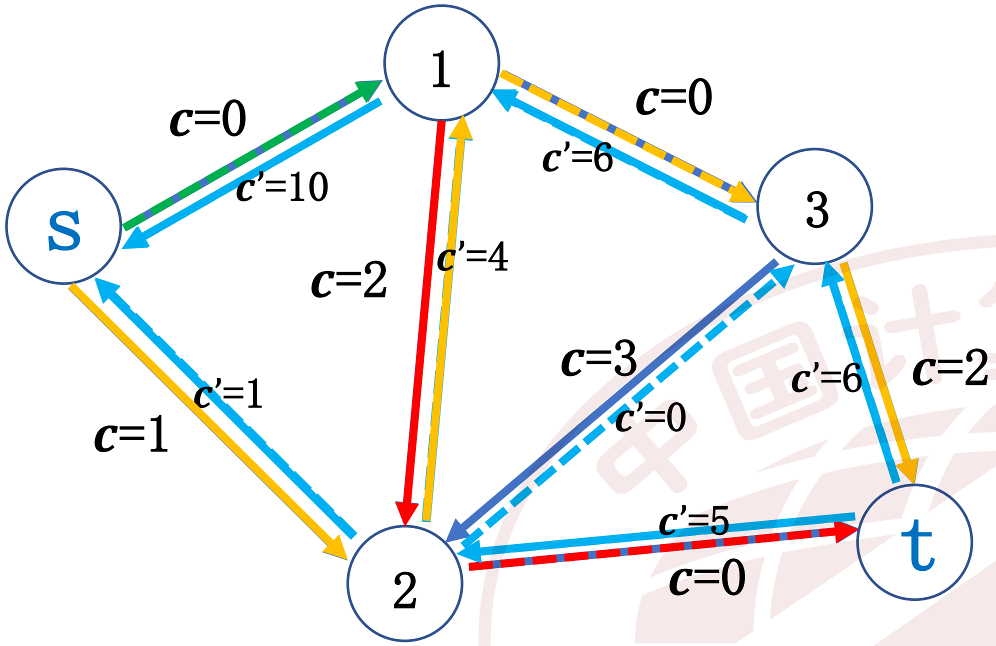
$$|f| = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in T}\sum_{v\in S} f(u,v)$$

$\because$ S 出边的总容量为上限，一定大于等于 S 的出边的总流量减去 S 的入边的总流量。

$\therefore$ $|f| \le c[S,T]$

$$|f| = f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) \le \sum_{u\in S}\sum_{v\in T} c(u,v) = c[S,T]$$

网络的最大流必定不超过最小割的容量。



最大流最小割定理 Ford-Fulkerson定理

1. f 是 G 的一个最大流
2. 残余网络 G_f 中不存在增广路
3. 对于G 的某个割 $[S,T]$，有$|f| = c[S,T]$

将 f 对应的残余网络 G_f 中，构造点集S：从源点s可以到达的顶点v 组成的集合

$\because$ 残余网络已经无法找到增广路，意味着无法达到汇点t

$\therefore$ $[S, V-S]$ 是 $s-t$割

$\because$ 割中都是满流边

$\therefore$ $|f| = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in T}\sum_{v\in S} f(u,v) = \sum_{u\in S}\sum_{v\in T} c(u,v) = c[S,T]$

$\therefore$ f 是最大流

- 如果所有边的容量都是整数，那么最大流和最小割必然也是整数。

最大流算法

Ford-Fulkerson方法

Edmond-Karp算法

Dinic算法

最大流-最小割定理

最大流算法的应用

二分图的匹配问题

上下界网络流

费用流算法

基于SPFA的实现

典型例题讲解

最小割模型的应用

典型例题讲解

最大权闭合图

二分图的最小点权覆盖集与

最大点权独立集

飞行员配对方案【网络流24题】

完整题面请访问 LibreOJ 6000
LuoGu P2756
AcWing 2175

第二次世界大战时期，英国皇家空军从沦陷国征募了大量外籍飞行员。

由皇家空军派出的每一架飞机都需要配备在航行技能和语言上能互相配合的 2 名飞行员，其中 1 名是英国飞行员，另 1 名是外籍飞行员。

在众多的飞行员中，每一名外籍飞行员都可以与其他若干名英国飞行员很好地配合。

对于给定的外籍飞行员与英国飞行员的配合情况，试设计一个算法找出最佳飞行员配对方案，使皇家空军一次能派出最多的飞机。

圆桌问题【网络流24题】

LibreOJ 6004
LuoGu P3254
完整题面请访问 AcWing 2179

假设有来自 m 个不同单位的代表参加一次国际会议。

每个单位的代表数分别为 $r_i\ (i = 1,2,\ldots,m)$。

会议餐厅共有 n 张餐桌，每张餐桌可容纳 $c_i\ (i = 1,2,\ldots,n)$ 个代表就餐。

为了使代表们充分交流，希望从同一个单位来的代表不在同一个餐桌就餐。

试设计一个算法，给出满足要求的代表就餐方案。

试题库问题【网络流24题】

中国计算机学会 China Computer Federation

完整题面请访问 LibreOJ 6006
LuoGu P2763
AcWing 2181

假设一个试题库中有 n 道试题。

每道试题都标明了所属类别，同一道题可能有多个类别属性。

现要从题库中抽取 m 道题组成试卷，并要求试卷包含指定类型的试题。

试设计一个满足要求的组卷算法。

餐饮【USACO 2007】

完整题面请访问 LuoGu P2891
AcWing 2240

每头奶牛都有自己喜欢的食物和饮料，并且不会食用其他不喜欢的食物和饮料。

农夫约翰为他的奶牛们做了美味的饭菜，但他忘了对照他们的喜好来检查菜单。

虽然他可能无法令所有奶牛满意，但他想给尽可能多的奶牛提供一顿完整的用餐：既有食物可吃，也有饮料可喝。

农夫约翰一共烹制了 F 种食物，并提供了 D 种饮料。

约翰共有 N 头奶牛，其中第 i 头奶牛有 F_i 种喜欢的食物以及 D_i 种喜欢的饮料。

约翰需要给每头奶牛分配一种食物和一种饮料，并使得有吃有喝的奶牛数量尽可能大。

每种食物或饮料都只有一份，所以只能分配给一头奶牛食用（即，一旦将第 2 种食物分配给了一头奶牛，就不能再分配给其他奶牛了）。

最长递增子序列问题【网络流24题】

完整题面请访问 LibreOJ 6005
LuoGu P2766
AcWing 2180

给定正整数序列 $x_1, \ldots, x_n$。

1. 计算其最长递增子序列的长度 s。
2. 计算从给定的序列中最多可取出多少个长度为 s 的递增子序列。给定序列中的每个元素最多只能被取出使用一次。
3. 如果允许在取出的序列中多次使用 x_1 和 x_n，则从给定序列中最多可取出多少个长度为 s 的递增子序列。

注意：递增指非严格递增。

最小路径覆盖问题【网络流24题】

完整题面请访问 LibreOJ 6002
LuoGu P2764
AcWing 2177

给定有向图 $G=(V,E)$。

设 P 是 G 的一个简单路（顶点不相交）的集合。

如果 V 中每个顶点恰好在 P 的一条路上，则称 P 是 G 的一个路径覆盖。

P 中路径可以从 V 的任何一个顶点开始，长度也是任意的，特别地，可以为 0。

G 的最小路径覆盖是 G 的所含路径条数最少的路径覆盖。

设计一个有效算法求一个有向无环图 G 的最小路径覆盖。

魔术球问题【网络流24题】

完整题面请访问 LibreOJ 6003
LuoGu P2765
AcWing 2178

假设有 n 根柱子，现要按下述规则在这 n 根柱子中依次放入编号为 $1,2,3,\ldots$ 的球。

每次只能在某根柱子的最上面放球。

在同一根柱子中，任何 2 个相邻球的编号之和为完全平方数。

试设计一个算法，计算出在 n 根柱子上最多能放多少个球。

例如，在 4 根柱子上最多可放 11 个球。

星际转移问题【网络流24题】

LibreOJ 6015
LuoGu P2754
完整题面请访问 AcWing 2187

由于人类对自然资源的消耗，人们意识到大约在 2300 年之后，地球就不能再居住了。于是在月球上建立了新的绿地，以便在需要时移民。令人意想不到的是，2077 年冬由于未知的原因，地球环境发生了连锁崩溃，人类必须在最短的时间内迁往月球。

现有 n 个太空站（编号 $1 \sim n$）位于地球与月球之间，且有 m 艘公共交通太空船在其间来回穿梭。每个太空站可容纳无限多的人，而每艘太空船 i 只可容纳 $H[i]$ 个人。

每艘太空船将周期性地停靠一系列的太空站，例如：$(1,3,4)$ 表示该太空船将周期性地停靠太空站 $134134134\ldots$。

每一艘太空船从一个太空站驶往任一太空站耗时均为 1 天。

人们只能在太空船停靠太空站(或月球、地球)时上、下船。

初始时所有人全在地球上，太空船全在初始站，即行驶周期中的第一个站。

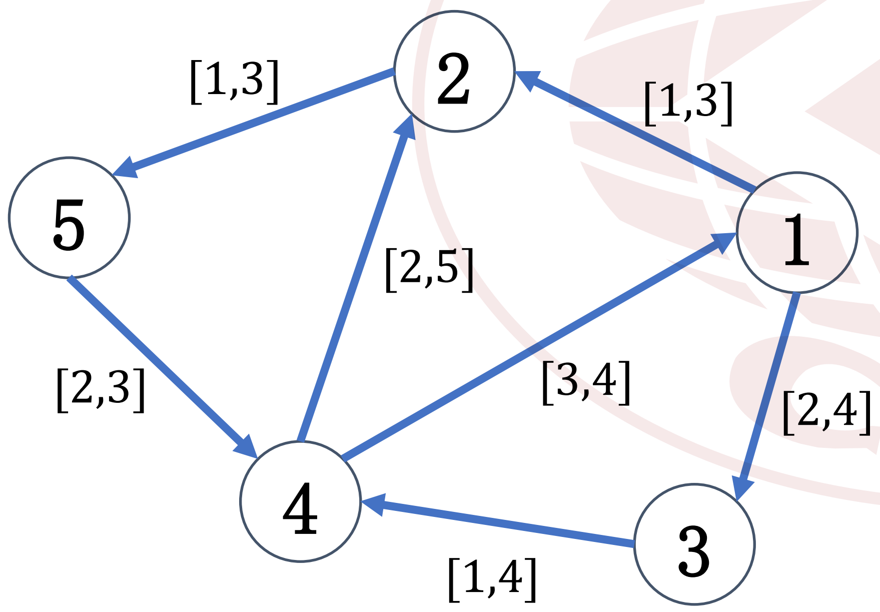
试设计一个算法，找出让所有人尽快地全部转移到月球上的最短天数。

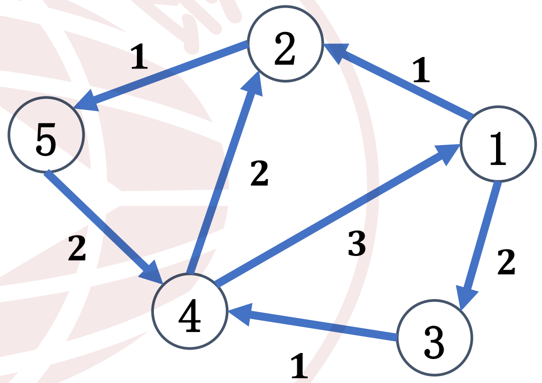
无源汇上下界可行流

LibreOJ 115
完整题面请访问 AcWing 2188

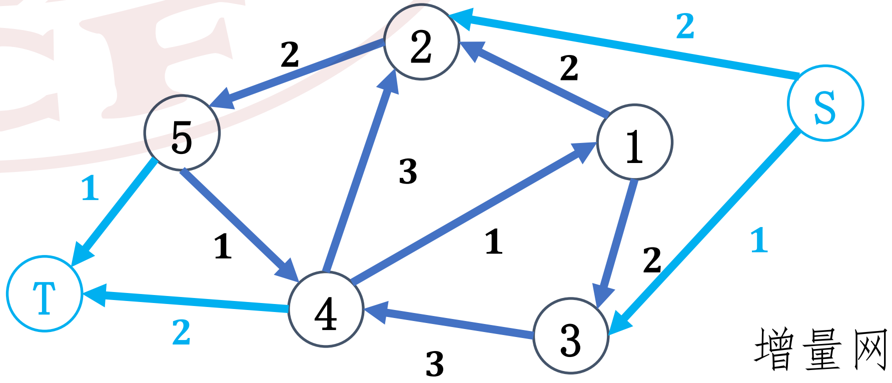
给定一个包含 n 个点 m 条边的有向图，每条边都有一个流量下界和流量上界。

求一种可行方案使得在所有点满足流量平衡条件的前提下，所有边满足流量限制。





低保网络



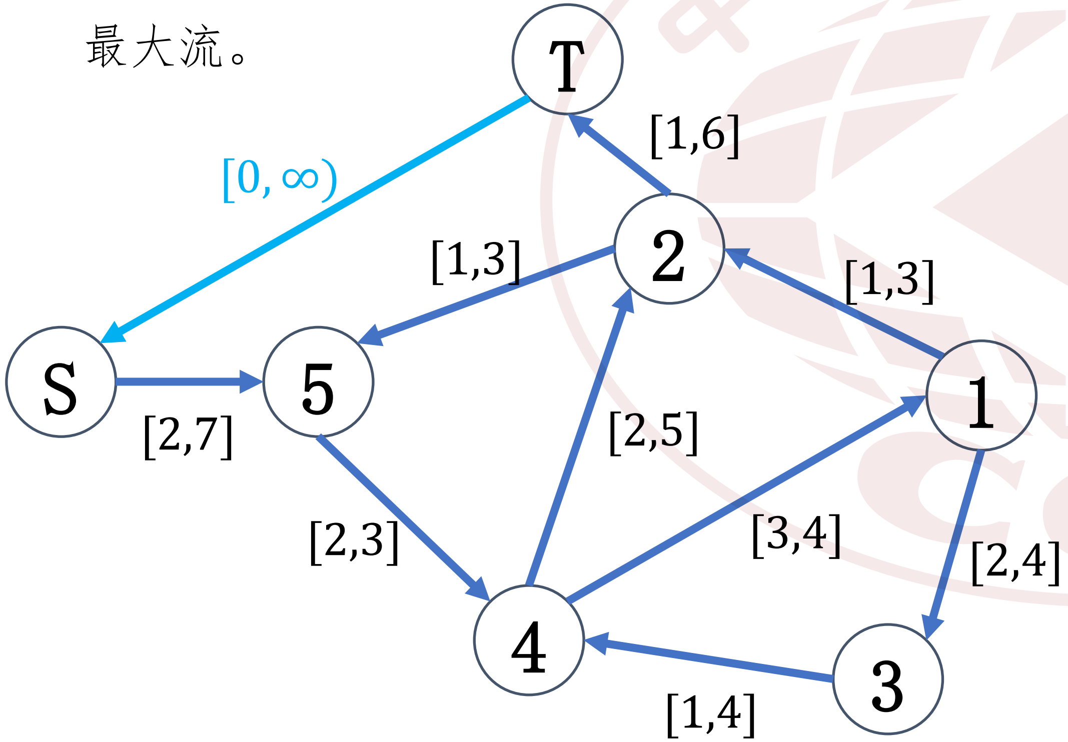
增量网络

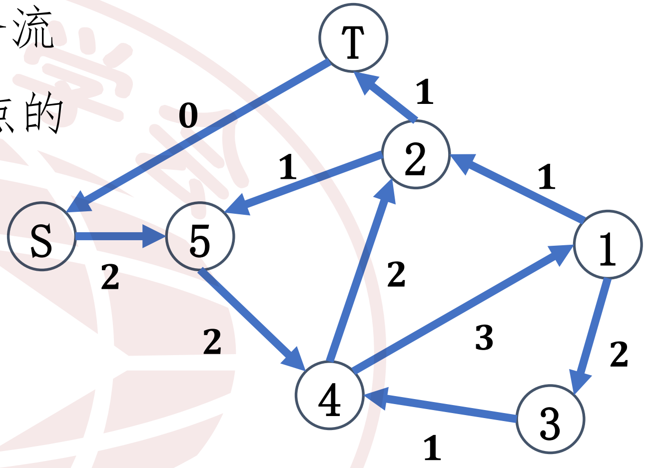
中国计算机学会 China Computer Federation

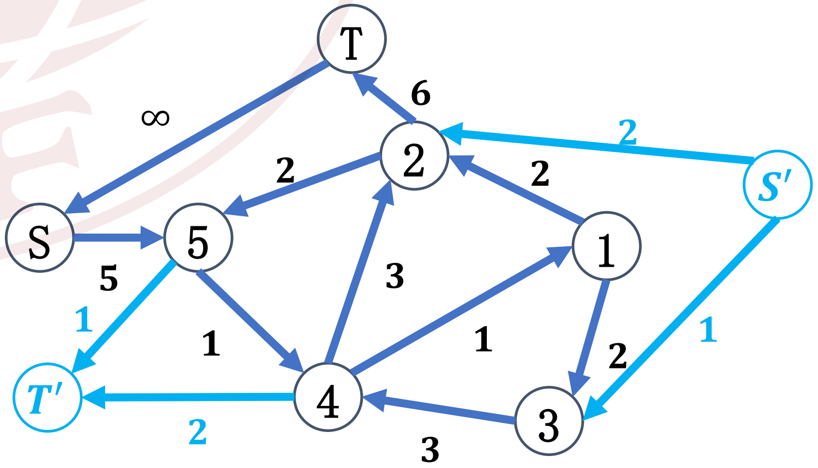
有源汇上下界最大流

LibreOJ 115
LuoGu P5192
完整题面请访问 AcWing 2189

给定一个包含 n 个点 m 条边的有向图，每条边都有一个流量下界和流量上界。给定源点 S 和汇点 T，求源点到汇点的最大流。







有源汇上下界最小流

完整题面请访问 LibreOJ 117
AcWing 2190

给定一个包含 n 个点 m 条边的有向图，每条边都有一个流量下界和流量上界。

给定源点 S 和汇点 T，求源点到汇点的最小流。

最大流算法

Ford-Fulkerson方法
Edmond-Karp算法
Dinic算法
最大流-最小割定理

最大流算法的应用

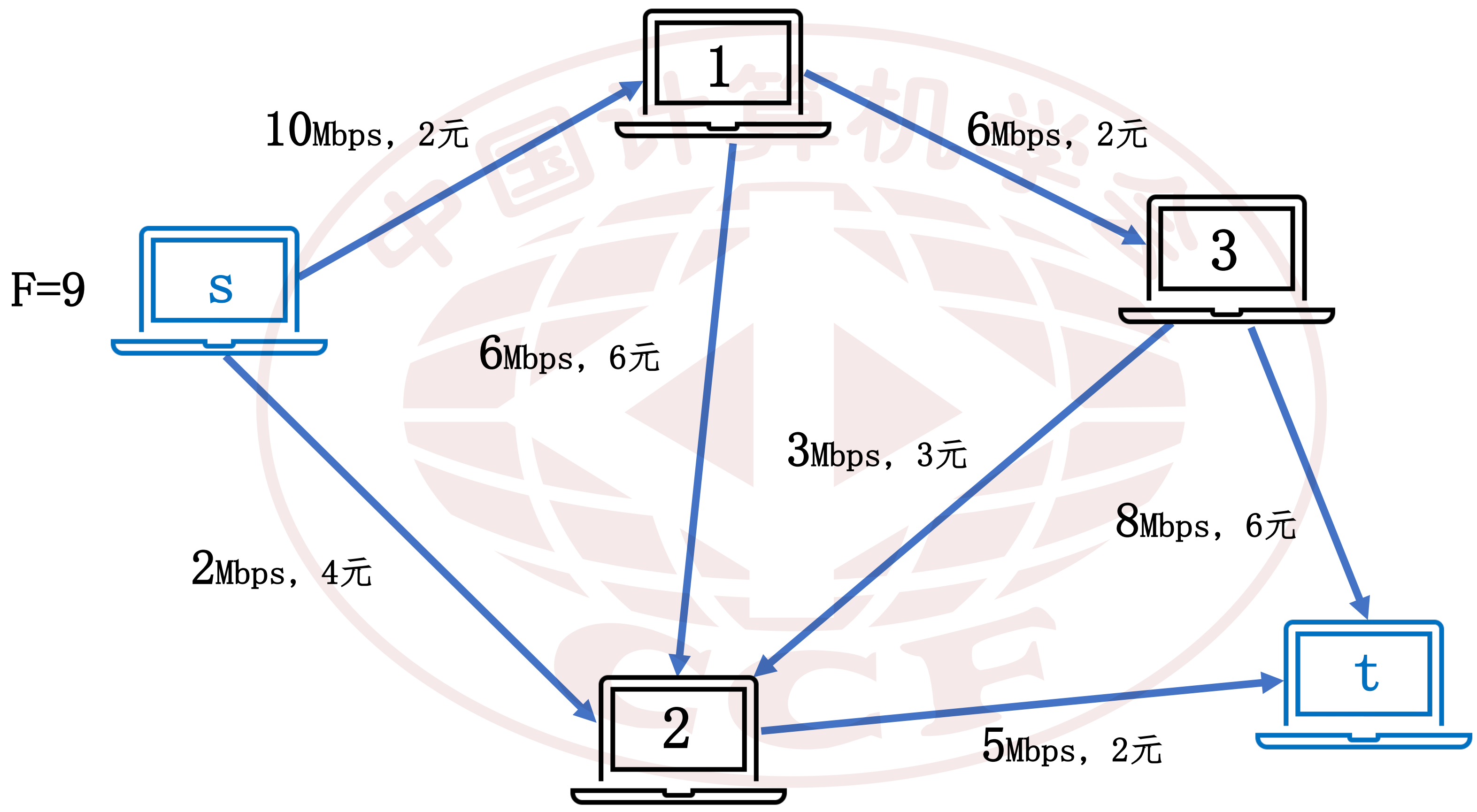
二分图的匹配问题
上下界网络流

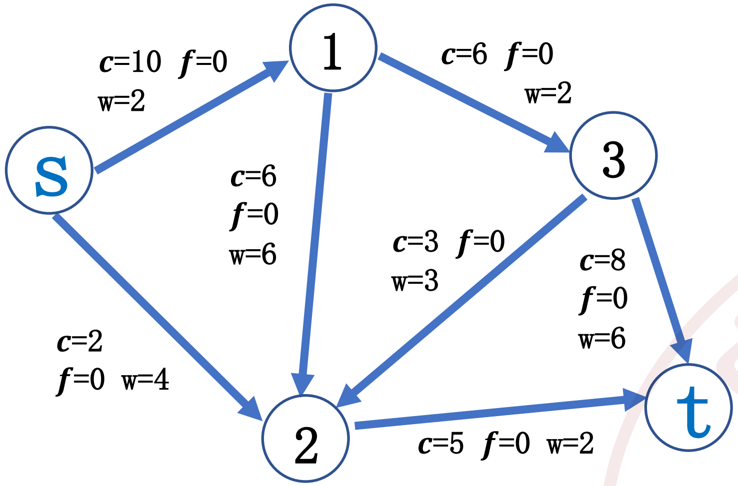
费用流算法

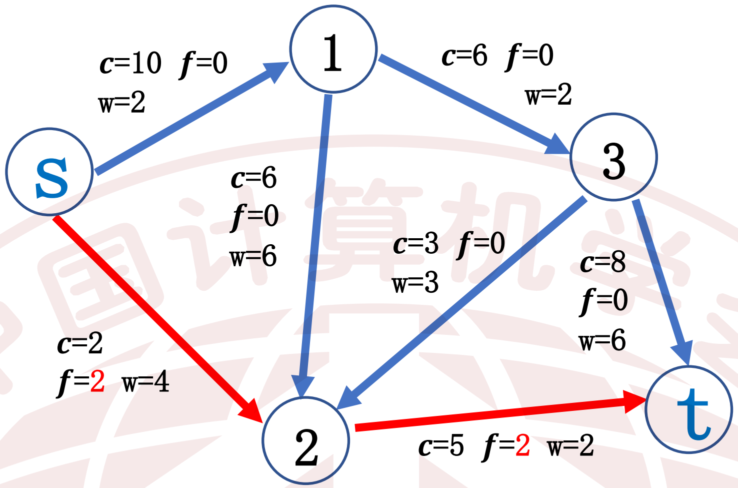
基于SPFA的实现
典型例题讲解

最小割模型的应用

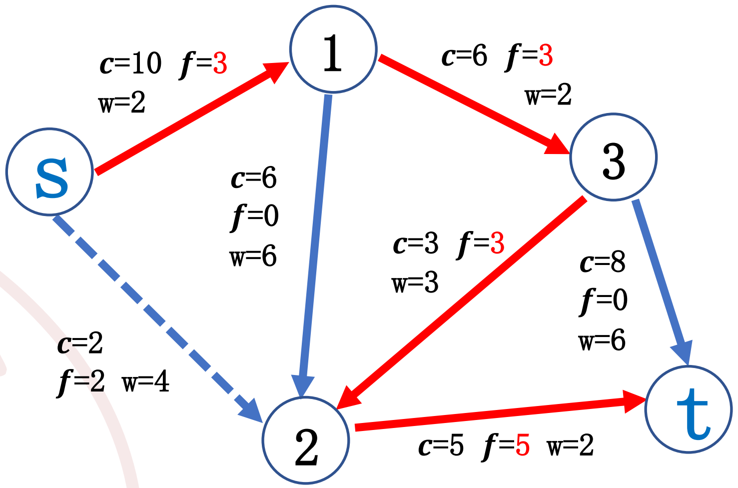
典型例题讲解
最大权闭合图
二分图的最小点权覆盖集与最大点权独立集


F=9
s
1
2
3
t
10Mbps，2元
6Mbps，2元
6Mbps，6元
3Mbps，3元
8Mbps，6元
2Mbps，4元
5Mbps，2元

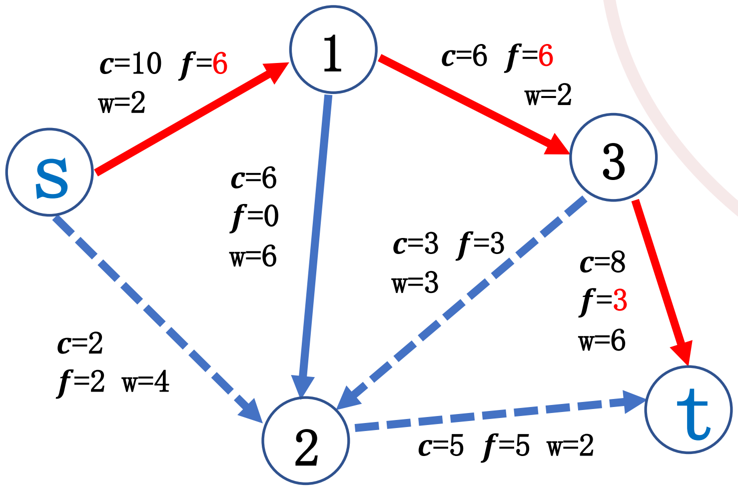




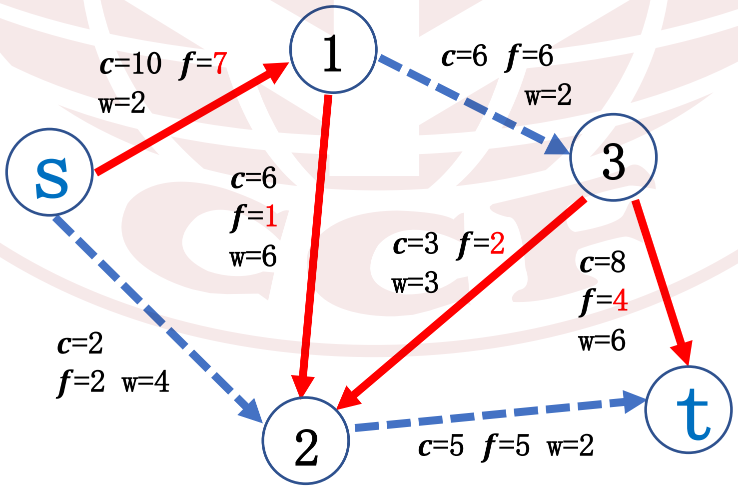
$$cost = (4+2)\times 2 = 12$$



$$cost = (2+2+3+2)\times 3 = 27$$



$$cost = (2+2+6)\times 3 = 30$$



$$cost = (2+6-3+6)\times 1 = 11$$

网络传输流量为 9 ，费用为 80

当流网络的边上增加了费用之后，

可以在残余网络上沿着最短路增广，

反向边的费用是正向边的值取反，

由于流网络中出现负权边，

需要使用Bellman-Ford（或 SPFA）算法。

运输问题【网络流24题】

LibreOJ 6011
LuoGu P4015
完整题面请访问 AcWing 2192

W 公司有 m 个仓库和 n 个零售商店。

第 i 个仓库有 a_i 个单位的货物；第 j 个零售商店需要 b_j 个单位的货物。

货物供需平衡，即 $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$。

从第 i 个仓库运送每单位货物到第 j 个零售商店的费用为 c_{ij}。

试设计一个将仓库中所有货物运送到零售商店的运输方案。

对于给定的 m 个仓库和 n 个零售商店间运送货物的费用，计算最优运输方案和最差运输方案。

负载平衡问题【网络流24题】

完整题面请访问 LibreOJ 6013
LuoGu P4016
AcWing 2194

G 公司有 n 个沿铁路运输线环形排列的仓库，每个仓库存储的货物数量不等。

如何用最少搬运量可以使 n 个仓库的库存数量相同。

搬运货物时，只能在相邻的仓库之间搬运。

航空路线问题【网络流24题】

LibreOJ 6122
LuoGu　P2770
完整题面请访问　AcWing　2185

给定一张航空图，图中顶点代表城市，边代表 2 城市间的直通航线。

现要求找出一条满足下述限制条件的且途经城市最多的旅行路线。

从最西端城市出发，单向从西向东途经若干城市到达最东端城市，然后再单向从东向西飞回起点(可途经若干城市)。

城市名出现的顺序是从西向东。也就是说，设 i,j 是城市表列中城市出现的顺序，当 $i>j$ 时，表示城市 i 在城市 j 的东边，而且不会有 2 个城市在同一条经线上。

除起点城市外，任何城市只能访问 1 次。

对于给定的航空图，试设计一个算法找出一条满足要求的最佳航空旅行路线。

分配问题【网络流24题】

LibreOJ 6012
LuoGu P4014
完整题面请访问 AcWing 2193

有 n 件工作要分配给 n 个人做。

第 i 个人做第 j 件工作产生的效益为 c_{ij}。

试设计一个将 n 件工作分配给 n 个人做的分配方案。

对于给定的 n 件工作和 n 个人，计算最优分配方案和最差分配方案。

数字梯形问题【网络流24题】

LibreOJ 6010
LuoGu P4013
完整题面请访问 AcWing 2191

给定一个由 n 行数字组成的数字梯形如右图所示。

梯形的第一行有 m 个数字。

从梯形的顶部的 m 个数字开始，在每个数字处可以沿左下或右下方向移动，形成一条从梯形的顶至底的路径。

```
      2   3
    3   4   5
  9  10   9   1
 1  1  10   1   1
1  1  10  12  1   1
```

规则 1：从梯形的顶至底的 m 条路径互不相交。

规则 2：从梯形的顶至底的 m 条路径仅在数字结点处相交。

规则 3：从梯形的顶至底的 m 条路径允许在数字结点相交或边相交。

对于给定的数字梯形，分别按照规则 1，规则 2，和规则 3 计算出从梯形的顶至底的 m 条路径，使这 m 条路径经过的数字总和最大。

最长k可重区间集问题【网络流24题】

LibreOJ 6014
LuoGu P3358
完整题面请访问 AcWing 2196

给定实直线 L 上 n 个开区间组成的集合 I，和一个正整数 k 。

试设计一个算法，从开区间集合 I 中选取出开区间集合 $S \subseteq I$，使得在实直线 L 的任何一点 x，S 中包含点 x 的开区间个数不超过 k，且 $\sum_{z \in S} |z|$ 达到最大。

这样的集合 S 称为开区间集合 I 的最长 k 可重区间集。

$\sum_{z \in S} |z|$ 称为最长 k 可重区间集的长度。

对于给定的开区间集合 I 和正整数 k，计算开区间集合 I 的最长 k 可重区间集的长度。

最长k可重线段集问题【网络流24题】

LibreOJ 6227
LuoGu P3357
完整题面请访问 AcWing 2197

给定平面 xoy 上 n 个开线段组成的集合 I，和一个正整数 k 。

试设计一个算法，从开线段集合 I 中选取出开线段集合 $S \subseteq I$，使得在 x 轴上的任何一点 p，S 中与直线 $x = p$ 相交的开线段个数不超过 k，且 $\sum_{z \in S} |z|$ 达到最大。

这样的集合 S 称为开线段集合 I 的最长 k 可重线段集。

$\sum_{z \in S} |z|$ 称为最长 k 可重线段集的长度。

对于任何开线段 z，设其断点坐标为 (x_0, y_0) 和 (x_1, y_1)，则开线段 z 的长度 $|z|$ 定义为：

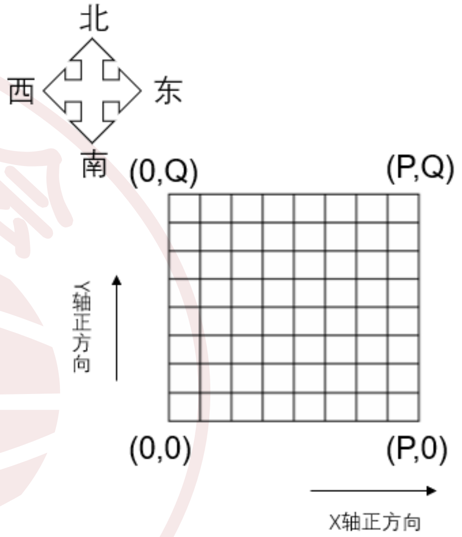
$|z| = \left\lfloor \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \right\rfloor$。

对于给定的开线段集合 I 和正整数 k，计算开线段集合 I 的最长 k 可重线段集的长度。

深海机器人问题【网络流24题】

完整题面请访问 LibreOJ 6224
LuoGu P4012
AcWing 2195

深海资源考察探险队的潜艇将到达深海的海底进行科学考察。潜艇内有多个深海机器人。潜艇到达深海海底后，深海机器人将离开潜艇向预定目标移动。深海机器人在移动中还必须沿途采集海底生物标本。沿途生物标本由最先遇到它的深海机器人完成采集。每条预定路径上的生物标本的价值是已知的，而且生物标本只能被采集一次。



深海机器人只能从其出发位置沿着向北或向东的方向移动，而且多个深海机器人可以在同一时间占据同一位置。

用一个 $P \times Q$ 网格表示深海机器人的可移动位置。西南角的坐标为 $(0,0)$，东北角的坐标为 (P,Q)。

给定每个深海机器人的出发位置和目标位置，以及每条网格边上生物标本的价值。

计算深海机器人的最优移动方案，使深海机器人到达目的地后，采集到的生物标本的总价值最高。

火星探险问题【网络流24题】

LibreOJ 6225
LuoGu P3356
完整题面请访问 AcWing 2198

火星探险队的登陆舱将在火星表面着陆，登陆舱内有多部障碍物探测车。登陆舱着陆后，探测车将离开登陆舱向先期到达的传送器方向移动。探测车在移动中还必须采集岩石标本。每一块岩石标本由最先遇到它的探测车完成采集。每块岩石标本只能被采集一次。岩石标本被采集后，其他探测车可以从原来岩石标本所在处通过。探测车不能通过有障碍的地面。

限定探测车只能从登陆处沿着向南或向东的方向朝传送器移动，而且多个探测车可以在同一时间占据同一位置。如果某个探测车在到达传送器以前不能继续前进，则该车所采集的岩石标本将全部损失。

用一个 $P \times Q$ 网格表示登陆舱与传送器之间的位置。登陆舱的位置在 (x_1, y_1) 处，传送器的位置在 (x_P, y_Q)处。

给定每个位置的状态，计算探测车的最优移动方案，使得所有探测车都成功到达传送器，而且探测车采集到的岩石标本的数量最多。保证登陆舱与传送器之间至少存在一条连通道路。

餐巾计划问题【网络流24题】

完整题面请访问 LibreOJ 6008
LuoGu P1251
AcWing 2184

一个餐厅在相继的 N 天里，每天需用的餐巾数不尽相同。

假设第 i 天需要 r_i 块餐巾 $(i = 1,2,\dots,N)$。

餐厅可以购买新的餐巾，每块餐巾的费用为 p 分；或者把旧餐巾送到快洗部，洗一块需 m 天，其费用为 f 分；或者送到慢洗部，洗一块需 n 天，其费用为 s 分。

餐厅每天使用的餐巾必须是今天刚购买的，或者是今天刚洗好的，且必须恰好提供 r_i 块毛巾，不能多也不能少。

每天结束时，餐厅必须决定将多少块脏的餐巾送到快洗部，多少块餐巾送到慢洗部，以及多少块保存起来延期送洗。

但是每天洗好的餐巾和购买的新餐巾数之和，要满足当天的需求量。

试设计一个算法为餐厅合理地安排好 N 天中餐巾使用计划，使总的花费最小。

最大流算法

Ford-Fulkerson方法
Edmond-Karp算法
Dinic算法
最大流-最小割定理

最大流算法的应用

二分图的匹配问题
上下界网络流

费用流算法

基于SPFA的实现
典型例题讲解

最小割模型的应用

典型例题讲解
最大权闭合图
二分图的最小点权覆盖集与最大点权独立集

有线电视网络

完整题面请访问 LuoGu UVA1660
AcWing 381

给定一张 n 个点 m 条边的无向图，求最少去掉多少个点，可以使图不连通。

如果不管去掉多少个点，都无法使原图不连通，则直接返回 n。

太空飞行计划问题【网络流24题】

LibreOJ 6001
LuoGu P2762
完整题面请访问 AcWing 2176

W 教授正在为国家航天中心计划一系列的太空飞行。

每次太空飞行可进行一系列商业性实验而获取利润。

现已确定了一个可供选择的实验集合 $E=\{E_1,E_2,\ldots,E_m\}$ 和进行这些实验需要使用的全部仪器的集合 $I=\{I_1,I_2,\ldots,I_n\}$。

实验 E_j 需要用到的仪器是 I 的子集 $R_j \subseteq I$。配置仪器 I_k 的费用为 C_k 美元。

实验 E_j 的赞助商已同意为该实验结果支付 P_j美元。

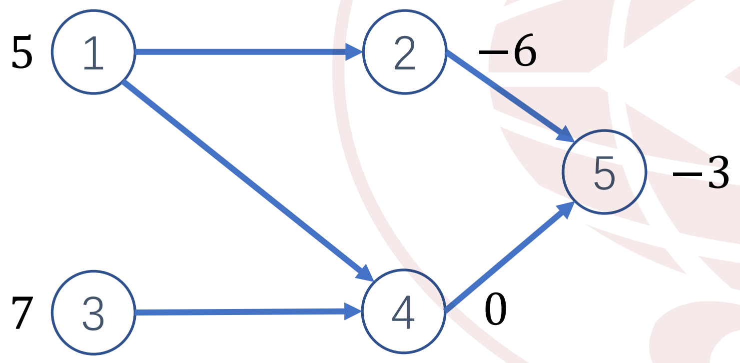
W 教授的任务是找出一个有效算法，确定在一次太空飞行中要进行哪些实验并因此而配置哪些仪器才能使太空飞行的净收益最大。

这里净收益是指进行实验所获得的全部收入与配置仪器的全部费用的差额。

对于给定的实验和仪器配置情况，找出净收益最大的试验计划。

最大权闭合图

- **闭合图**：是有向图 $G=(V,E)$ 的一个点集，且该点集的所有出边都还指向该点集，即闭合图内的任意点的任意后继也一定在闭合图中。
- **最大权闭合图**：（如给 G 中每个点 v 分配点权 w_v）点权之和最大的闭合图，即最大化$\sum_{v\in V'} w_v$。



左图中的闭合图共有9个：

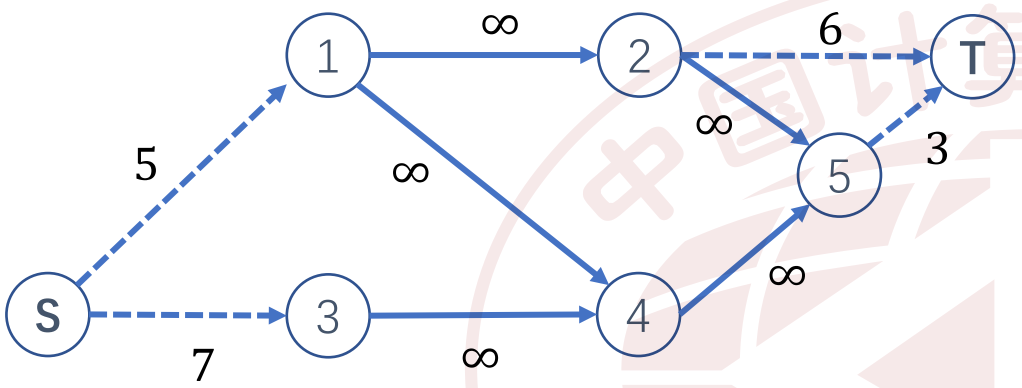
$\emptyset, \{5\}, \{2,5\}, \{4,5\}, \{2,4,5\}, \{3,4,5\},$

$\{1,2,4,5\}, \{2,3,4,5\}, \{1,2,3,4,5\}$

最大权闭合图为 $\{3,4,5\}$，权和为 4

- 闭合图的性质反映了事件间的必要条件的关系：一个事件的发生，它所需要的所有前提也都要发生。最大权闭合图对应了获益最大或效率最高的事件选择集合。

将最大权闭合图转化为流网络



$V_N = V \cup \{s, t\}$

$$\begin{cases} c(u,v) = \infty & (u,v) \in E \\ c(s,v) = w_v & w_v > 0 \\ c(v,t) = -w_v & w_v < 0 \end{cases}$$

$E_N = E \cup \{(s,v) | v \in V, w_v > 0\} \cup \{(v,t) | v \in V, w_v < 0\}$

- **简单割**：满足割中的每条边都只与源 s 或汇 t 关联的 $s-t$ 割。
- 在最大权闭合图转化的网络中，最小割是简单割。
- 约定：设简单割 $[S,T]$ 将网络 N 的点集 V_N 划分为点集 S 和补集 $T(T = V_N - S)$，满足 $s \in S, t \in T$。设闭合图为 V_1，它在 V 中的补集为 $V_2(V_2 = V - V_1)$。记 V^+ 为 V 中点权为正的点集，V^- 为 V 中点权为负的点集。同样的，可以定义 V_1^+ 与 V_1^-，V_2^+ 与 V_2^- 。

（**可行性**）网络 N 的简单割 $[S,T]$ 与图 G 的闭合图方案存在一个一一对应的关系：$V_1 \cup \{s\} = S$；

（**对应关系的数量化**）在一一对应关系下，有：$c[S,T] = \sum_{v \in V_2^+} w_v + \sum_{v \in V_1^+} (-w_v)$

（**最优性**）当网络 N的简单割取到最小割时，其对应的图 G 的闭合图将达到最大权，即：$w(V_1) = \sum_{v \in V^+} w_v - c[S,T]$

太空飞行计划问题【网络流24题】

LibreOJ 6001
LuoGu P2762
完整题面请访问 AcWing 2176

W 教授正在为国家航天中心计划一系列的太空飞行。

每次太空飞行可进行一系列商业性实验而获取利润。

现已确定了一个可供选择的实验集合 $E=\{E_1,E_2,\ldots,E_m\}$ 和进行这些实验需要使用的全部仪器的集合 $I=\{I_1,I_2,\ldots,I_n\}$。

实验 E_j 需要用到的仪器是 I 的子集 $R_j \subseteq I$。配置仪器 I_k 的费用为 C_k 美元。

实验 E_j 的赞助商已同意为该实验结果支付 P_j美元。

W 教授的任务是找出一个有效算法，确定在一次太空飞行中要进行哪些实验并因此而配置哪些仪器才能使太空飞行的净收益最大。

这里净收益是指进行实验所获得的全部收入与配置仪器的全部费用的差额。

对于给定的实验和仪器配置情况，找出净收益最大的试验计划。

方格取数问题【网络流24题】

中国计算机学会
China Computer Federation

LibreOJ 6007
LuoGu P2774
完整题面请访问 AcWing 2183

在一个有 $m \times n$ 个方格的棋盘中，每个方格中有一个正整数。

现要从方格中取数，使任意 2 个数所在方格没有公共边，且取出的数的总和最大。

试设计一个满足要求的取数算法。

对于给定的方格棋盘，按照取数要求编程求出最大的总和。

图的一些概念

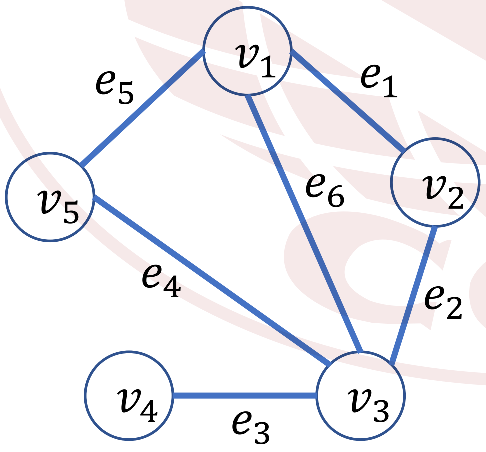
- 匹配：在图中两两没有公共端点的边集合。
- 边覆盖集： 图中任意顶点都至少是此集合中某条边的端点。
- 点覆盖集： 图中的任意边都有至少一个端点属于此集合。
- 点独立集：在图中两两互不相连的顶点集合。

最大匹配 $\{e_1, e_3\}, \{e_2, e_5\}$

最小边覆盖 $\{e_1, e_3, e_4\}$

最小点覆盖集$\{v_1, v_3\}$

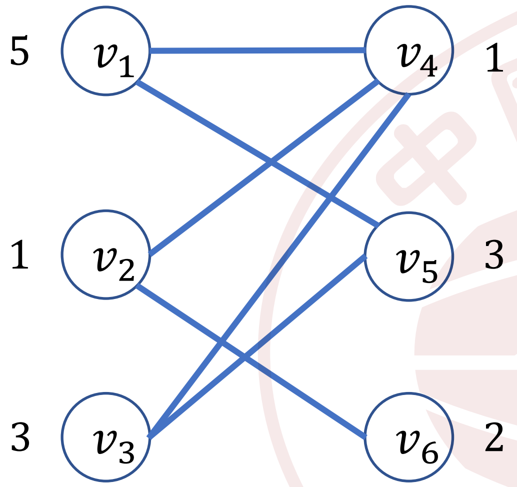
最大点独立集$\{v_2, v_4, v_5\}$

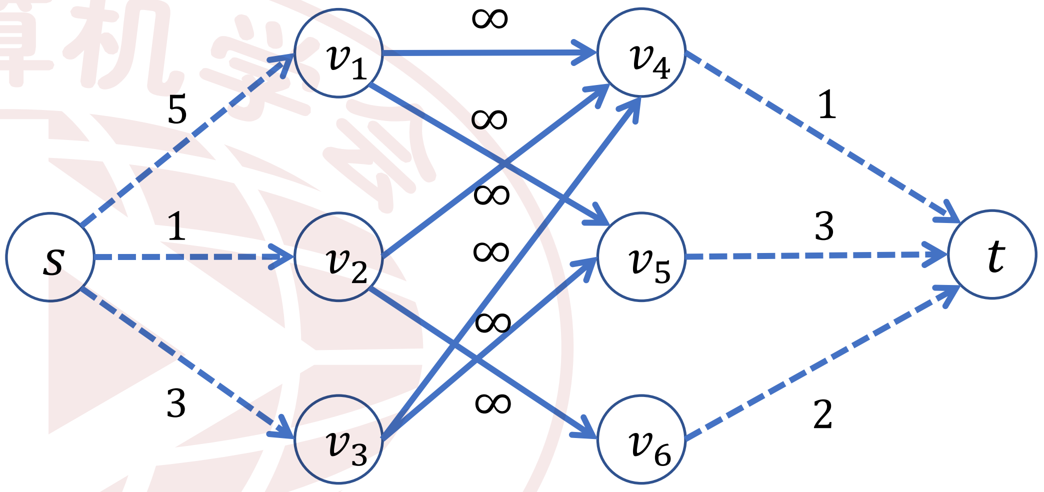


对于不存在孤立点的图，最大匹配数+最小边覆盖数=$|V|$

一个带点权二分图$G = (V, E)$中，其中$V = X \cup Y, X \cap Y = \emptyset$，且对于$\forall v \in V$，都被分配了一个非负点权 $w_v (w_v \geq 0)$。在该二分图中，研究最小点权覆盖集和最大点权独立集。

二分图的最小点权覆盖集算法





网络 N 的简单割 $[S,T]$ ，与图 G 的点覆盖集存在一一对应关系：点覆盖集中的点在网络 N 中相应的带权边组成一个简单割，反之亦然。

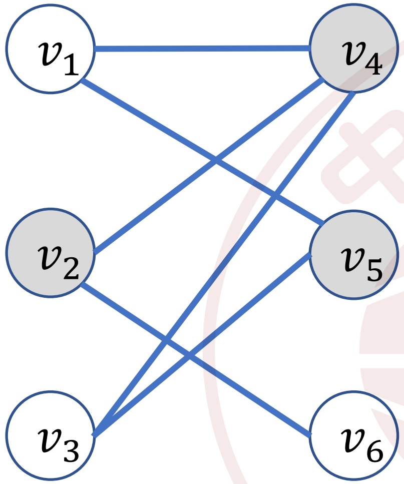
因为两者一一对应，且最小割与最小权点覆盖的优化方向一致，所以本问题可以用最小割解决。

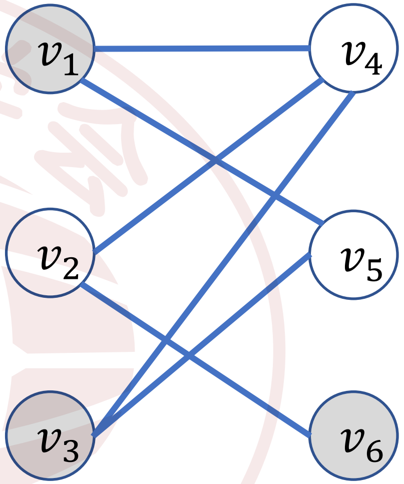
$$V_N = V \cup \{s,t\}$$

$$E_N = E \cup \{(s,u) | u \in X\} \cup \{(v,t) | v \in Y\}$$

$$\begin{cases} c(u,v) = \infty & (u,v) \in E \\ c(s,u) = w_u & u \in X \\ c(v,t) = w_v & v \in Y \end{cases}$$

二分图的最大点权独立集算法





（覆盖集与独立集互补定理）在一个不含孤立点的任意图中，若 V_1 是图的一个点覆盖集，当且仅当 它的补集是该图的一个点独立集。

（最优性）在一个不含孤立点的任意图中，若 V_1是图的一个最小点覆盖集，则它的补集是该图的一个最大点独立集。

方格取数问题【网络流24题】

LibreOJ 6007
LuoGu P2774
完整题面请访问 AcWing 2183

在一个有 $m \times n$ 个方格的棋盘中，每个方格中有一个正整数。

现要从方格中取数，使任意 2 个数所在方格没有公共边，且取出的数的总和最大。

试设计一个满足要求的取数算法。

对于给定的方格棋盘，按照取数要求编程求出最大的总和。

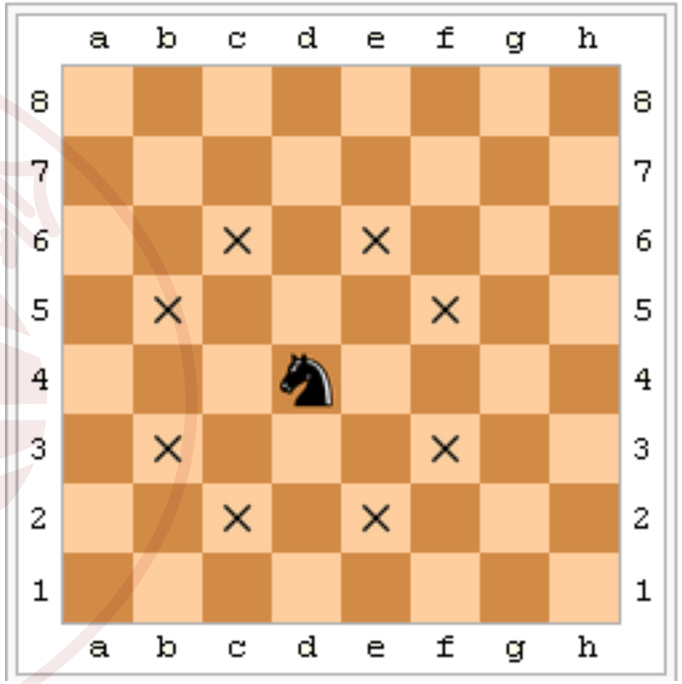
骑士共存问题【网络流24题】

LibreOJ 6226
LuoGu P3355
完整题面请访问 AcWing 2199

在一个 $n*n$ 个方格的国际象棋棋盘上，马（骑士）可以攻击的棋盘方格如图所示。

棋盘上某些方格设置了障碍，骑士不得进入。

对于给定的 $n*n$ 个方格的国际象棋棋盘和障碍标志，计算棋盘上最多可以放置多少个骑士，使得它们彼此互不攻击。



最小割模型应用总结

- 最小割是最大流这一直观问题的对偶问题，但更抽象。
- 转化过程的模式
  - 建立原问题与最小割的一一对应关系是化归方法的必要步骤。
  - 将原问题的决策方案与具有特定性质的割，构造性的一一对应起来，并且保证具有该特定性质的割一定能被最小割取到。
- 割的性质
  - （不连通）在给定的流网络中，去掉割的边集，则不存在任何一条从源到汇的路径
  - （两类点）在给定的流网络中，任意一个割将点集划分为两部分，割为两部分点之间的桥梁。
- 技巧
  - 用“正无限容量”排除不参与决策的边。
  - 使用割的定义式来分析最优性。
  - 利用与源或汇关联的边容量处理点权。

参考资料

- 秋叶拓哉，《挑战程序设计竞赛》
- 李煜东，《算法竞赛进阶指南》
- 胡伯涛，《最小割模型在信息学竞赛中的应用》
- “网络流24题”题解参考： Beyond the Void
- 程序参考：AcWing、LibreOJ

T H A N K S