网络流
Network Flow

2017-8-10

内容列表

- 基本知识点清单
- 特殊网络
- 限制条件与对应模型（2016，2017）
- 其它性质（2017）

基本知识点清单

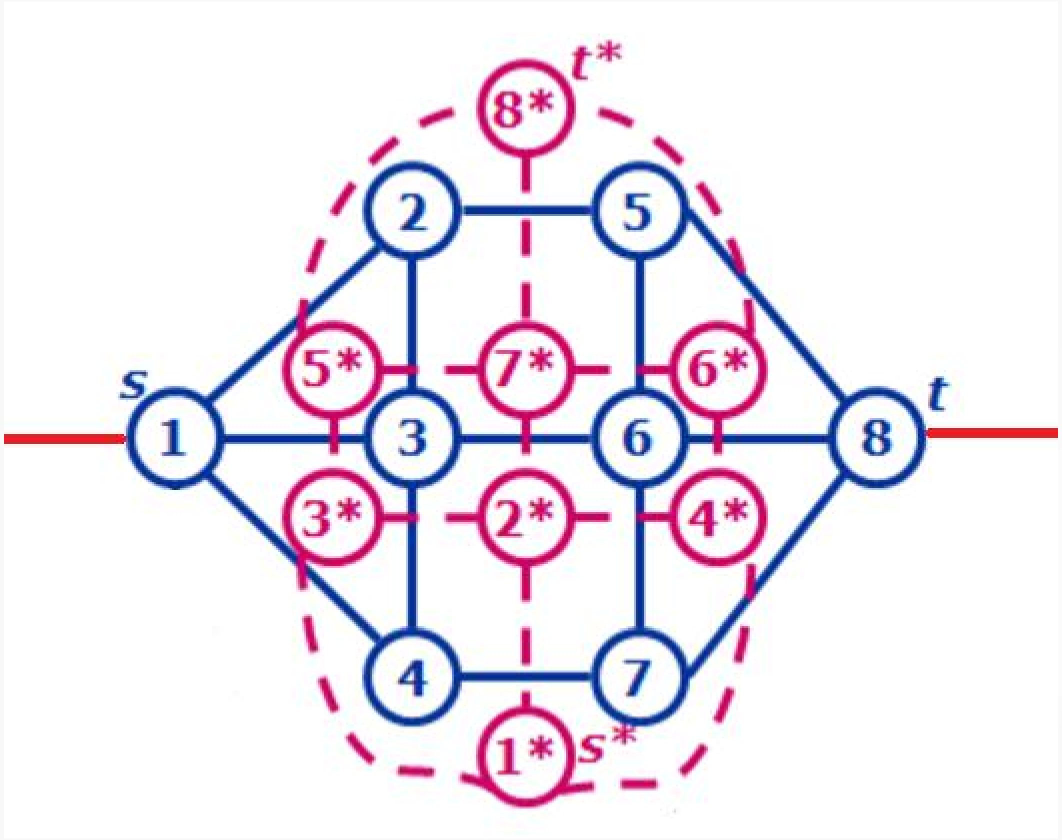
- 1）二分图最大匹配（匈牙利算法，Dinic/Hopcraft Karp）
  - 找可行增广路
  - 找一条哈密顿回路，找一条欧拉回路
- 2）二分图最优匹配（KM，费用流）
- 3）最大流
  - 每次增广一条路径（EK，DFS，BFS）
  - Dinic（为什么推荐Dinic，一些优化？）
- 4）费用流（最小费用流 与 最小费用最大流）
  - 每次增广一条路径（找最短路径）
  - 找负环
- 5）特殊限制
  - 带上下界
  - 费用函数不是线性函数
- 6）常见问题模型：网络流24题

特殊网络

- 平面图
- 二分图
- 小规模流量上界
- 单位图与单出度
- 特殊图结构

特殊网络 - 平面图

- 平面图最大流 ＝ 平面图最小割 ＝ 对偶图最短路



特殊网络 - 平面图

- 例题一
- 考虑 2*n 的网格图，求 (1,1) 到 (2,n) 的最小割

特殊网络 - 平面图

- 例题二

- 给定平面图G
- 之后有大量询问，对于询问点对(u,v)求最小u-v割

- Gomory-Hu tree

特殊网络 - 二分图

- 二分图最大（匹配）流 $O(M\ N^{1/2})$

- Hopcraft Karp
- Dinic

特殊网络 - 小规模流量上界

- 每一条边的流量限制都是整数
- 且最大为C

- O(NMlogC)
- 最高标号推进

特殊网络 - 单位图与单出度

- 直接Dinic

- $O(MN^{2/3})$
- $O(MN^{1/2})$

特殊网络 - 特殊图结构

- 链式图
- 分层图

特殊网络 - 链式图 - 产品销售(CTSC2010)

A 公司正在热销某计算机产品，作为 A 公司 CEO 的小 A 打算为接下来连续的 N 个销售季度制定一份具体的生产和销售方案。已知第 i 个销售季度该产品的订购量为 D_i，在第 i 个季度，A 公司会通过如下几种方式来解决用户的订购需求：

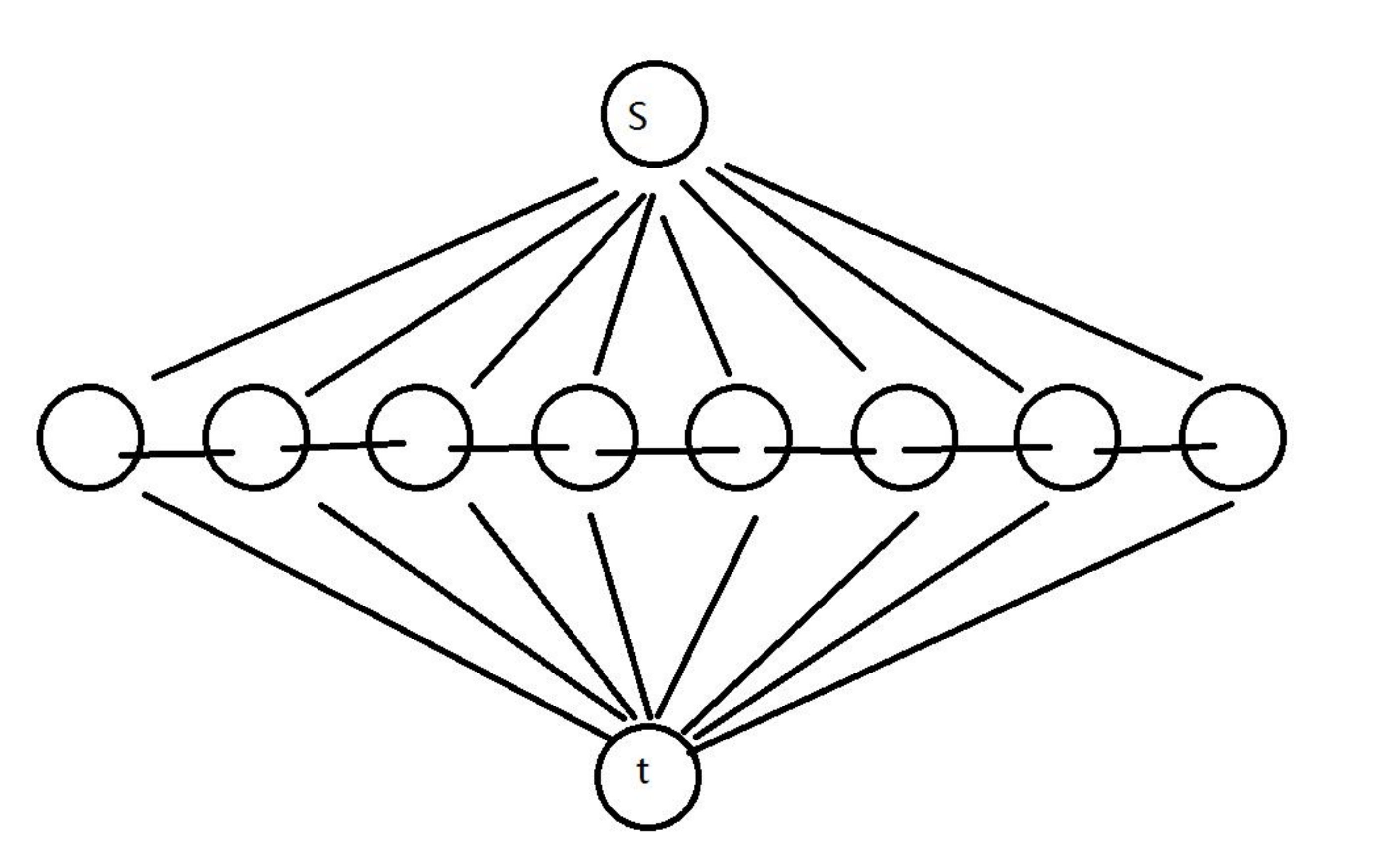
- 在第 i 个季度生产新的产品来销售。
- 若在第 i 个季度以前库存还有多余的产品，则可以直接在第 i 个季度销售(注意第一个季度之前没有任何库存产品)。
- 在第 i 个季度可以不完成全部的订购需求，而将未完成的订购需求推迟，归入到下一个季度($i+1$)的产品订购需求中。

A 公司需要考虑以下几种耗费：**生产新产品的成本耗费**、库存产品的**额外储存耗费**以及推迟订购需求而需要**赔偿给用户的损失费**。另外由于劳力和资源的限制，每个销售季度能够生产新产品的数量是有限的，各季度的耗费和可以生产的产品上限数也不尽相同，具体如下：

- 在第 i 个季度最多可以生产 U_i 件新的产品，每一件的成本为 P_i。
- 第 i 个季度保存下来的产品，可以用于以后季度的销售。对于每一件产品，若从第 i 季度保存到第 $i+1$ 季度，需要额外支付 M_i 的存储费用(注意产品保存到下个季度后可能再次库存)。
- 对于第 i 个季度需要推迟而归入到下一个季度订购需求的每一件产品，A 公司需要赔偿给用户损失费 C_i(注意延迟到下个季度可能再次被延迟，费用按后面季度的延迟费用计)。

在第 N 个季度结束后，A 公司必须解决之前所有的用户订单。可以保证，A 公司能够生产的产品总数不会低于总订购量，也就是说**一定存在一组生产和销售方案使得满足所有的用户订购需求**。小 A 想知道如何来安排产品的生产和销售，使得在满足所有订购需求的前提下公司总的耗费最小。

特殊网络 - 链式图



限制条件与对应模型

- 要求不可达 = 切断所有增广路
- 二选一 = 切左或切右
- 三选一 = 切左或切中或切右
- 强制依赖条件 = 与条件内所有点连边的特殊点
- = 强制过边
- 点限制 = 拆点后限制边
- 边流量大小限制 = 利用出度入度相等
- 路径限制
- 递增的分段费用（第一次免费）= 拆边
- 贪心与费用流
- 有优先级的限制 = 改变费用

要求不可达 = 切断所有增广路

- **Hold Your Hand**
- Fang Fang hates some 8-digit binary numbers.
- Cupid can sell me some supernatural powers.
- Some of them can eliminate all 8-digit binary numbers in the world with a certain prefix, and some of them can eliminate all 8-dight binary numbers with a certain suffix (offer ur IQ in exchange for them).
- I should minimize my damage.

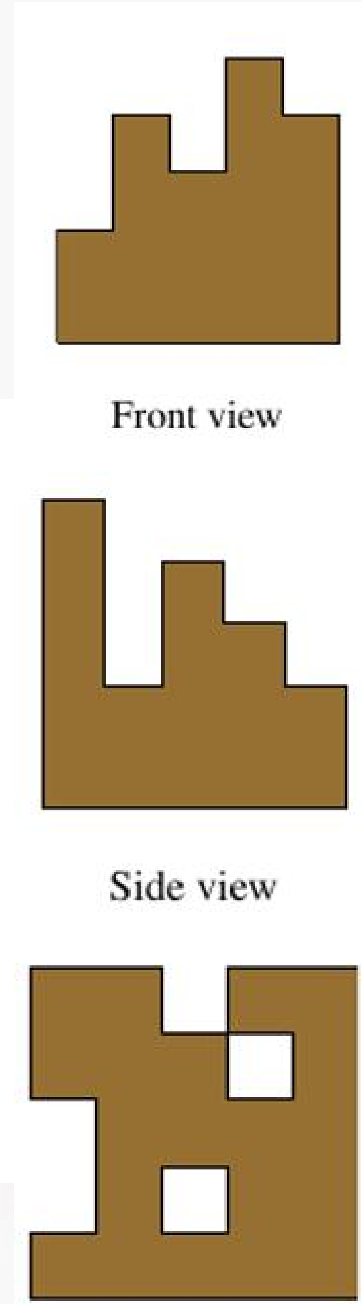
二选一 = 切左或切右

- **Mission Improbable**

Side camera →

1	4	0	5	2
2	1	2	0	1
0	2	3	4	4
0	3	0	3	1
1	2	2	1	1

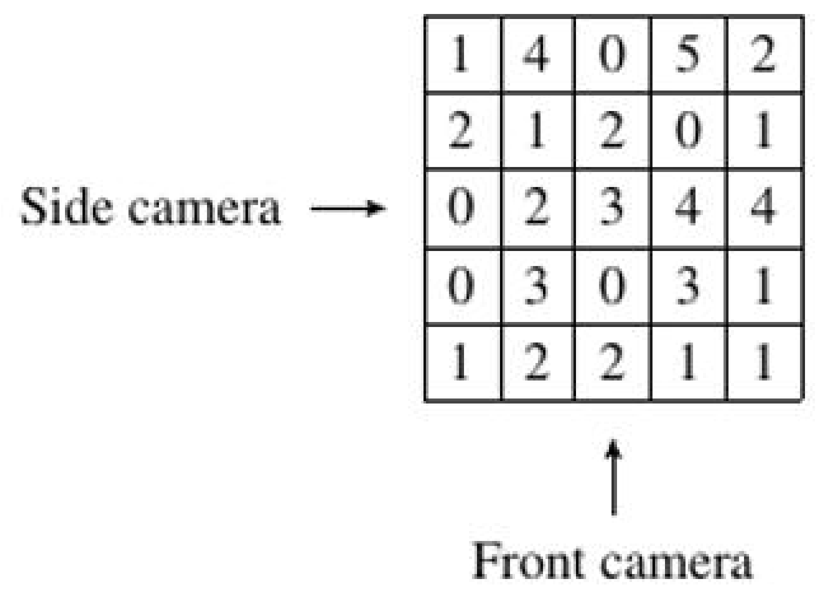
Front camera



- **！如果没有0怎么做？**

二选一 = 切左或切右

- **Mission Improbable**
- **按照高度从高到低**

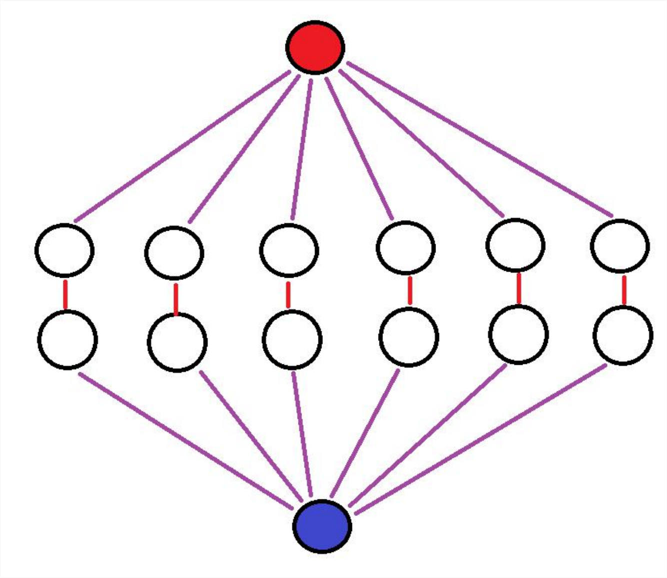


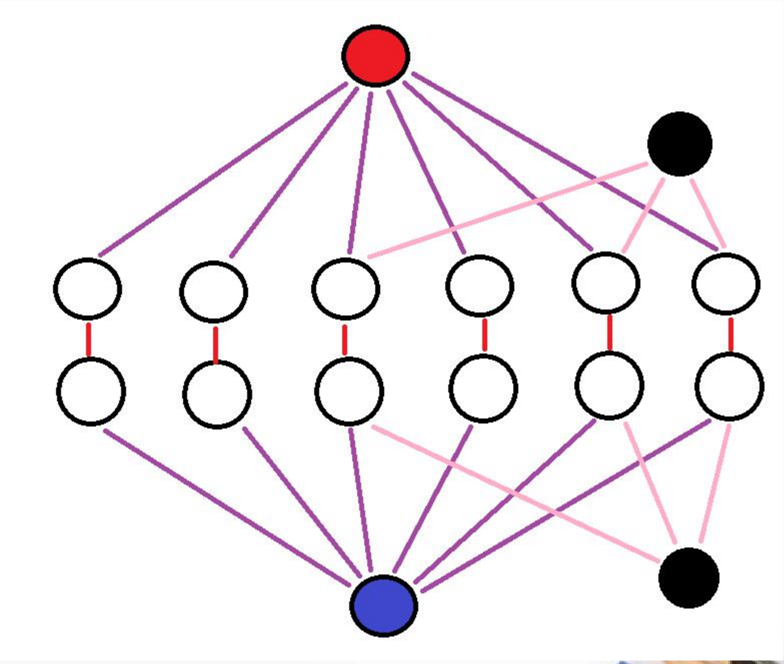
- **对于高度h（假设右图中4 5的格子都是0；h=3）**
  - **有2行的最大高度为3**
  - **有3列的最大高度为3**
  - **左2右3的二分图**
  - **(2,2) 是 0，不可能（不存在连边）**

强制依赖条件=与条件内所有点连边的特殊点

- **双语句子**
- 给定若干单词，其中有些是英语单词，一些是法语单词，也有可能同时是英语单词和法语单词
- 已知其中一些单词是英语单词，其中一些单词是法语单词
- **还给了若干句子（句子内的单词是同一种语言）**
- 最少有多少双语单词？

强制依赖条件=与条件内所有点连边的特殊点





点限制 = 拆点后限制边

- **Catering**

- **给定费用流模型**
- **要求找一个流量为K的最小费用流**
- **满足除了s t外每一个点都经过了恰好一次**

路径限制

- 例题一：路径固定为k的路径最大流

- 给定最大流模型
- 找一个最大流，满足所有从s到t的路径长度都恰好是k

路径限制

- **例题二：不超过3的路径最大流**

- **给定最大流模型**
- **找一个最大流，满足所有从s到t的路径长度都不超过3**

路径限制

- 例题二：不超过3的路径最大流

- 长度1：s到t的边 e=<s,t>
- 长度3的路径：分层图（分四层）
- 长度2的路径：
  - 枚举所有长度为2的路径
  - 拆中间点
  - 分层图（依然是四层）

一道综合题 - 芯片的难题

【问题描述】 一家著名的微处理器公司请您帮助他们在一些他们自己的电脑芯片上安装一些可互换组件（部件）。每块芯片都被设计成$N \times N$ 的带插槽的正方形。一个插槽可以安装一块单独的组件，你的任务是尽可能多地插入这些组件。

现代处理器的设计是很复杂的。你要面对下面几个限制：

- 一些插槽是不可用的。
- 一些插槽已经被其它的组件占据了，因而无法被新的组件使用。
- 内存总线要连接到芯片的水平和垂直的边界上，它们的带宽负载需要匹配。除外，在第一行和第一列上的组件数目必须一样多，在第二行和第二列上的组件数目也必须一样多，依此类推。计数的组件包括之前已经存在于芯片之上和后来加上去的。
- 类似地，由于一些电源连接到行尾和列尾。为了避免插槽过热，对于给出的A和B，在任何行和列上的组件总数不能多于芯片上组件总数的A/B。

一道综合题 - 芯片的难题

芯片被描述为一个N行，每行N个字符的矩阵，其中'.' 表示开放插槽，'/' 表示不可用插槽，'C' 表示插槽已被一个组件占据。举例来说：

```
CC/..
././/
..C.C
/.C..
/./C/
```

图8. 4-1

一道综合题 - 芯片的难题

如果每行每列上的组件不可以超过组件总数的3/10，那么这块5 ×5 的芯片的可被安装的最大组件数是7。图8. 4-2是一种可行方案，其中'W' 表示新加到开放插槽中的组件。

```
CC/W.
W/W//
W.C.C
/.CWW
/W/C/
```

输入：

输入数据由多组测试用例组成。每组测试用例由含有3 个整数的一行开始：芯片的规模N ($1\le N\le 40$)，以及A和B ($1\le B\le 1000$, $0\le A\le B$)，含义如上所述。然后给出N 行，每行N个字符描述插槽，字符为'.'，'/' 或'C' 之一，含义如上所述。

一道综合题 - **芯片的难题**

- 试题要求计算安装在开放插槽中的最多组件数，即安装的最多组件总数-必须放的组件数。
- 依次枚举安装的组件总数Answer
- （Answer的枚举范围？）

- 如果没有条件(3)：
  - 网格的每行和每列分别作为一个节点u_i和v_i
  - 构成二分图。

一道综合题 - **芯片的难题**

- 如果不考虑条件(2)
  - 可以用费用流来维护答案
- 加入条件(2)
  - 保证其一定被放置
  - 修改其费用为-INF
  - 最优解cost
    - [cost / INF] 是必须安装的组件数
    - [cost % INF] 是新安装的组建数

一道综合题 - **芯片的难题**

- 对于条件(3)
  - 满足第i行的组件个数=第i列的组件个数

- 构造环流
- 平衡流

其它性质

- 网络流问题本身更是一个图论问题

- 1）什么是残余网络
- 2）从最优化的角度看
- 3）一些额外增加的限制

- （还有更多的分析可能，见下午的习题）

什么是残余网络

- “残余”是对原网络说的
- 并不是这个网络有缺陷

- 不能把流网络思考为无向图！
- 正向弧 - 反向弧 必须同时存在

- 都是一个正常的流网络

什么是残余网络

- 都是一个正常的流网络
- 动态网络流维护（有向边流量上界变化）：
  - To Wax
    - 直接 capacity += 1
    - 依然是合法的残余网络：可以继续尝试增广
  - To Wane
    - 直接 capacity -= 1？不行！
    - 先退流：向s退1的流量
    - 向t找一条流量路径

从最优化的角度看

- 每一条边的流量看作变量 e
- 则：
  - 入流 = 出流，提供了大量恒等关系，消元
  - 每一条边有流量限制（对应为对变量的线性限制）
  - A convex hull !
- 重新整理定义
  - 变元 $e = (e_1,e_2,e_3,\ldots,e_k)$ （k是多少？）
  - 凸包 H = H(G)；最大流问题 f(H)
- 考虑 G' 有着和 G相同的网状结构
  - capacity in G' 严格不小于capacity in G
  - f(H(G')) >= f(H(G))
- f是凸函数：
  - 把 f 写作 capacities 的函数

一些额外添加的限制

- 记住：
  - 最大流问题（费用流）是某个凸包内的最优化问题
  - 最优化函数是凸函数
- 例题一：

  费用流

  额外给定边集 K subset E

  要求K集合内边流量相同

一些额外添加的限制

- 例题一：
  费用流
  额外给定边集 K subset E
  要求K集合内边流量相同

- 如何刻画这个额外的限制？
- 不妨画一个 k = 3, |K| = 2的例子！

- （类似动态规划在凸壳上找最值）

一些额外添加的限制

- 例题二：不超过6的路径最大流
- （不妨简化一点，先想想不超过4的路径怎么办？）

训练题列表

- Uvalive 5131 Chips Challenge
- Uvalive 6032 Minimum Cost Flow
- Uvalive 6395 Surely You Congest
- Uvalive 6778 Sensor Network
- Uvalive 7152 Catering
- Uvalive 8041 Mission Improbable

- **欢迎提问！**