数据结构和算法实习

郭 炜

学会程序和算法，走遍天下都不怕!

讲义照片均为郭炜拍摄

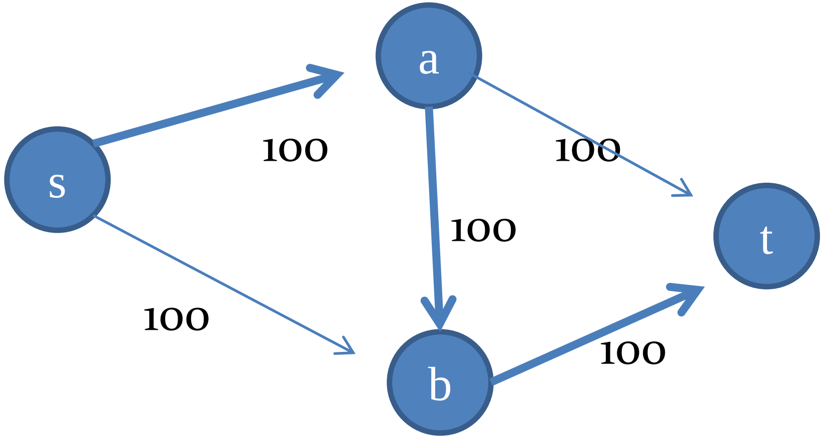
网络流问题

有流量下界的最大流问题

瑞士马特洪峰

有流量下界的网络最大流

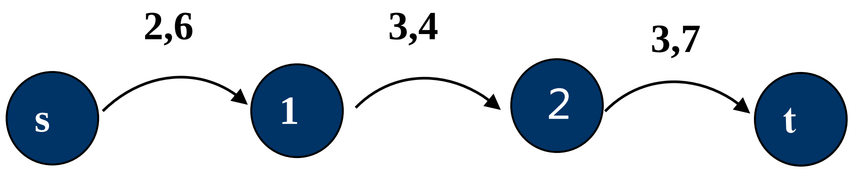
若规定a->b至少要有流量100，则最大流就是100，而不是200



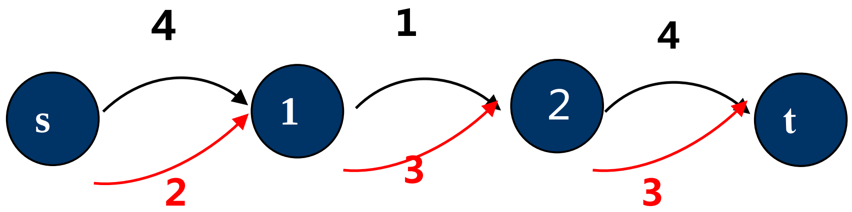
有流量下界的网络最大流

。

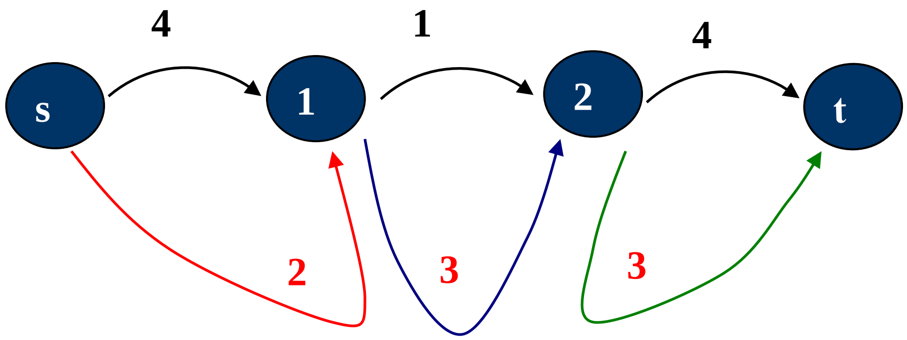
- 思路：将下界“分离”出去，使问题转换为下界为 0 的普通网络流问题。



将原边(u,v)分离出一条必要边和一条非必要边：
假设B(u,v)是下界，则分离出两条边：
C1(u,v) = B(u,v) —— 必要边，必须满流
C2(u,v) = C(u,v) - B(u,v) —— 非必要边

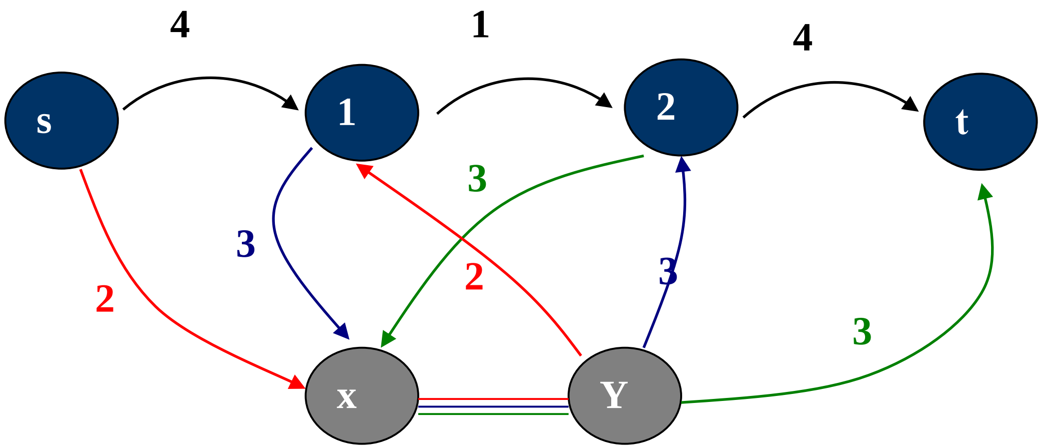


有流量下界的网络最大流



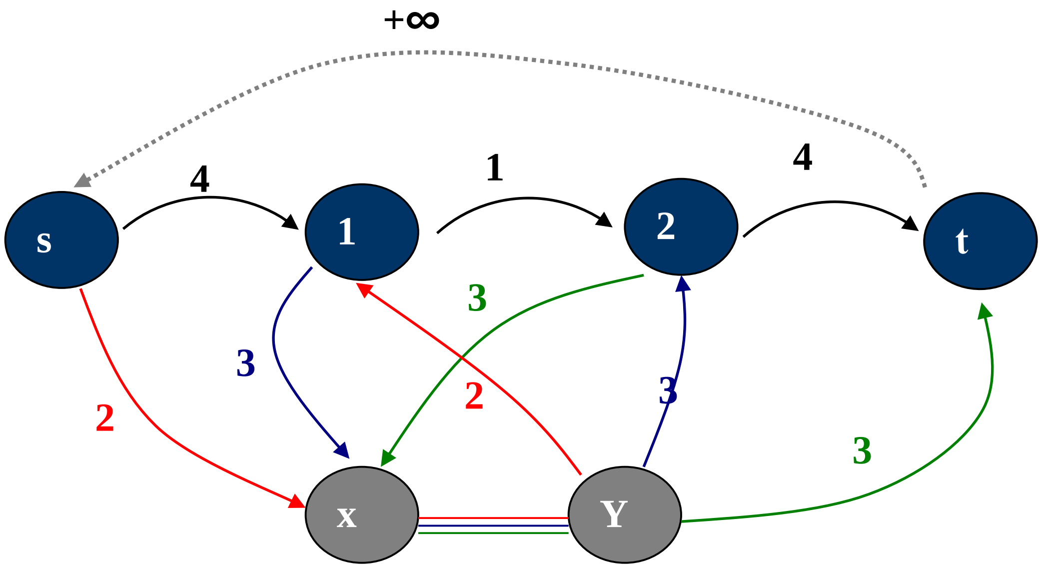
有流量下界的网络最大流

- 构造以下等价网络流图，每个必要边拆成两段，第一段->x->y->第二段
- x->y 的边容量无穷大



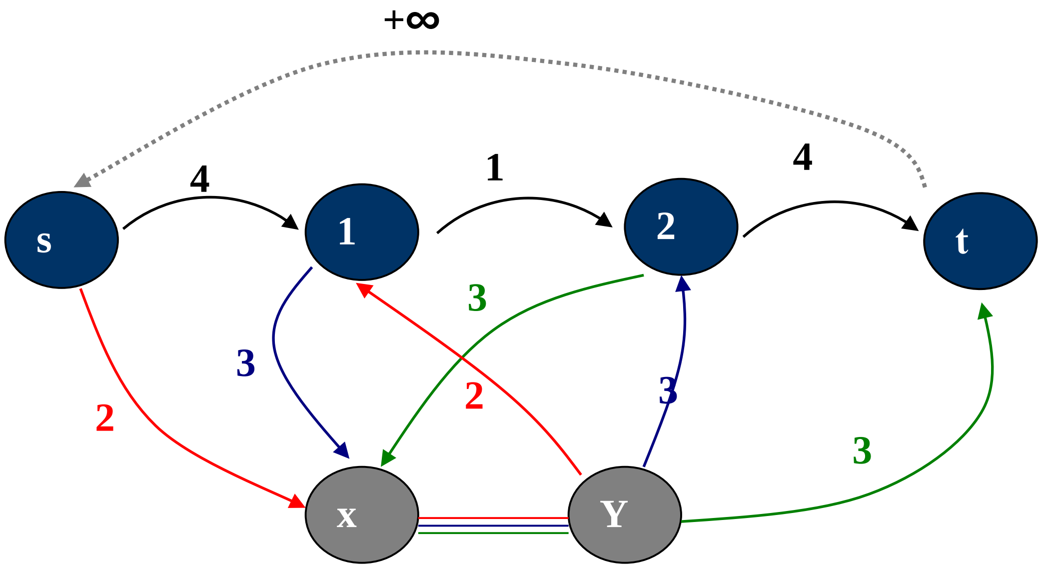
有流量下界的网络最大流

添加由t到s的容量为正无穷大的边，想象水可以从s流入t流出又流入s，在内部循环流动，且必要边都是满流。



有流量下界的网络最大流

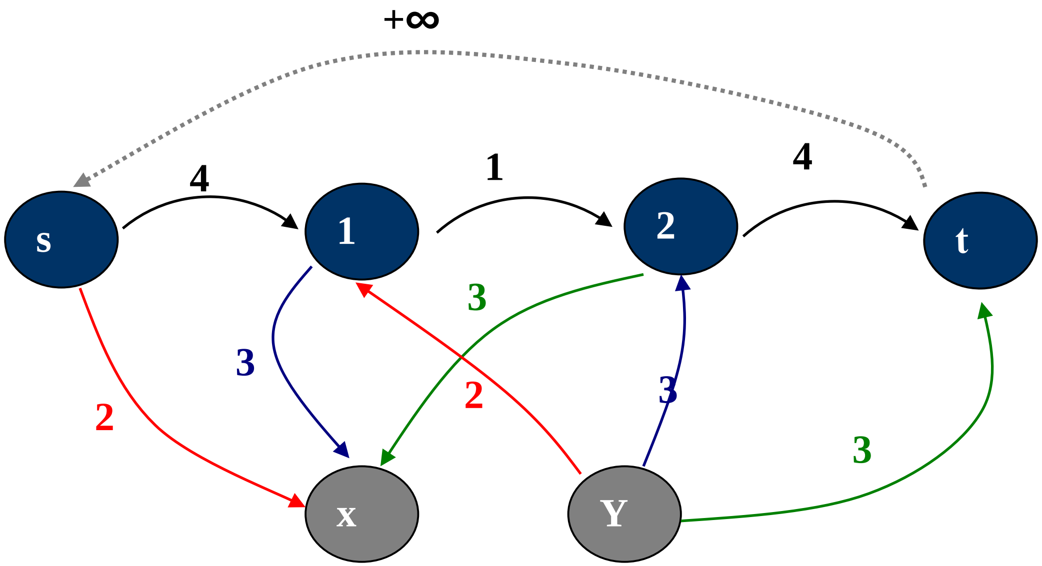
去掉边（x,y)，添加由t到s的容量为正无穷大的边，使y和x分别成为新的源和新的汇。



有流量下界的网络最大流

去掉边（x,y），添加由t到s的容量为正无穷大的边，使y和x分别成为新的源和新的汇。若此图上的最大流能够占满与Y相连的所有边的容量，则原图上就存在满足下界条件的可行流。若最大流不能够占满与Y相连的所有边的容量，则原图不存在可行流。

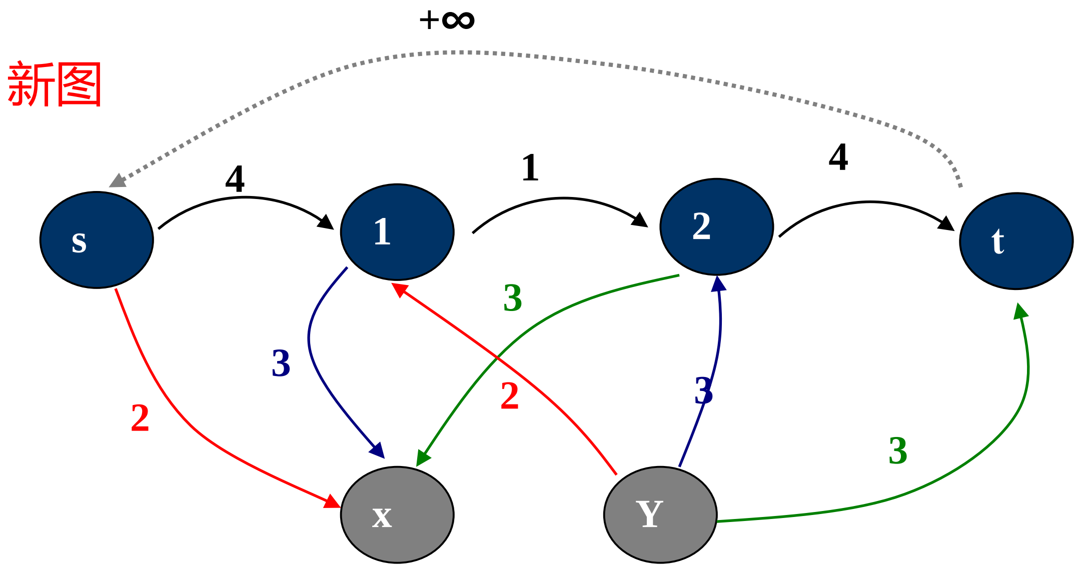
新图



有流量下界的网络最大流

新图最大流若小于新图中 y 的流入量之和,则原问题无解

在新图的最大流中，求出s流出的流量之和，记为sum1



有流量下界的网络最大流

- 在做过一遍最大流的新图的残余网络中，去掉t->s以及s->t的边，然后以s为源，t为汇再做一次最大流，此时得到的流量 sum2，则 sum1+sum2就是在原图上满足下界的最大流。
- 和x,y相连的边不用处理，因为x,y实际上是只能流入或只能流出的点，在图中不起作用。
- 要想求出每条边上的流量，怎么办?

有流量下界的网络最大流

- 要想求出每条边上的流量，怎么办?

- 在做第二次最大流之前，将新图的残余网络备份到G2
- 经过两次求最大流后，新图最后变成的残余网络是G
- 此时G2[i][j] – G[i][j] + LC[i][j] 就是 i->j上的流量
- LC[i][j] 是i->j边上的流量下界(下界是被满足的)

有流量下界的网络最大流

处理网络流题目要注意，如果有重边，则要将重边上的容量和下界累加，合并成一条边。

例题1
Budget

瑞士卢塞恩

例题1: POj 2396 Budget

一个n*m的矩阵, 矩阵里每个元素都是正整数，且满足以下限制条件:

1> 第i行的数和必须为 SH_i
2> 第i列的数和必须为 SL_i
3> 某些格子里的数，大小有限制。例如第2行第3列的数字必须大于5(或必须小于3,或必须等于10等)

求是否存在这样的矩阵。若有，输出该矩阵。无则输出IMPOSSIBLE.

例题1: POj 2396 Budget

- 每行看作一个节点，编号从1……n
- 每列看作一个节点，编号从n+1……n+m
- 添加源点 s = 0 和 汇点 t = n+m+1.

1> 将源点和每一个行节点连边，容量和下界都设为该行所有数的和
2> 将每一个列节点和汇点连边， 容量和下界都设为该列所有数的和
3> 如果u行v列的数字必须大于w,则边<u,v+n>流量的下界是w+1
4> 如果u行v列的数字必须小于w,则边<u,v+n>容量为w-1
5> 如果u行v列的数字必须等于w,则边<u,v+n>流量的下界和容量都是w
6> 其它情况，从每个行节点到每个列节点连边，容量为无穷大

- 找到的可行流（必然就是最大流），就是问题的解

例题1: POj 2396 Budget

本题trick:

1) W可能为负数，产生流量下界为负数的情况。应处理成0

2) 数据本身可能矛盾。比如前面说了 (2,1) =1,后面又说(2,1) = 10

最小费用最大流

列支敦士登

最小费用最大流

若网络流图中的每条边都有一个在此边的单位流量所需的费用f（简称边的费用)，则可引入最小费用最大流问题:

在所有最大流中，找一个总费用最小的（最大流可能不唯一）。

$$\min_{f \in F} a(f) = \min_{f \in F} \sum_{(i,j) \in E} a(i,j) f(i,j)$$

其中a代表流量，F为最大流的集合，即在最大流中寻找一个费用最小的最大流。

最小费用最大流

- 反复用spfa算法做源到汇的最短路进行增广，边权值为边上单位费用。反向边上的单位费用是负的。最短路就是总费用最少的路。
- 直到无法增广，即为找到最小费用最大流。
- 成立原因：每次增广时，每增加1个流量，所增加的费用都是最小的(贪心算法)
- 因为有负权边，所以不能用迪杰斯特拉算法求最短路。

例题2
Farm Tour

古罗马斗兽场

例题2: POJ 2135 Farm Tour

有n个景点，一个人要从1号景点走到n号景点，再从n号景点走到1号。要求回来的路不能重复(和去的路不能有公共边，可以有公共点，不一定走完所有景点，只要求从1到n即可。已知一些景点之间的路的长度（双向），问最短需要走多少路才能回来?

例题2: POJ 2135 Farm Tour

- 由于去和回来可以看成：2条从1到n的不同的路。所以转化成求从1到n的两条不同的路，让两条路的总长度最短
- 把人看作流，把每条边看作只能让一个人通过（容量为1），则问题变成要让两个人从1流到n，能否成功(即最大流可否为2)。若能，还要求总路程最短的方案。
- 令每条边的费用都是边的长度，总路程最短即是费用最小。问题变成最小费用最大流。

例题2: POJ 2135 Farm Tour

建网络流图：

- 建立源点，连接1号景点，费用0，容量 2 （表示可以有2个人走)
- 建立汇点，连接n号景点，费用0 ，容量2
- 若原图边（a,b）长度为 C，则网络流图上：
  边(a,b) 费用为C，容量是1，边(b,a) 费用为C，容量是1。
- 若最大流是2，就表示了有两条从1到n的不同的路
- 最小费用最大流的最小费用就是最短路径长度。

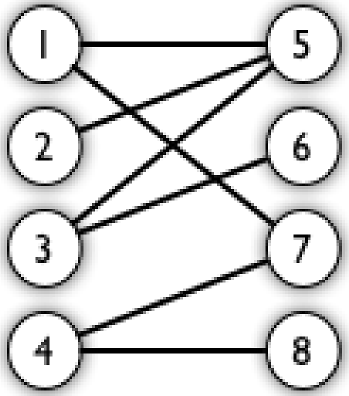
二部图的最大匹配

威尼斯

二部图的最大匹配

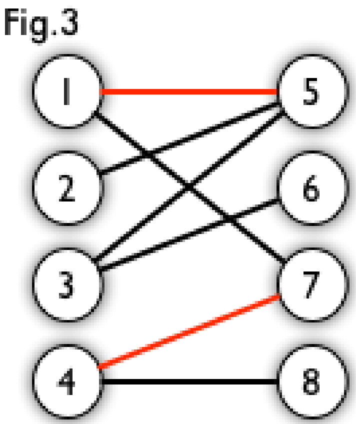
二分图(二部图)：如果能把一个图的顶点划分为两个不相交集 U 和 V，使得每一条边都分别连接U 、 V 中的顶点，则则此图为一个二分图。

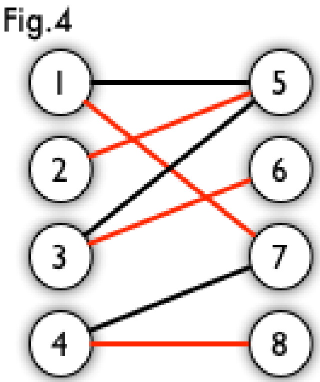
Fig.2



二部图的最大匹配

匹配：是一个边的集合，其中任意两条边都没有公共顶点。例如，图 3、图 4 中红色的边就是匹配。

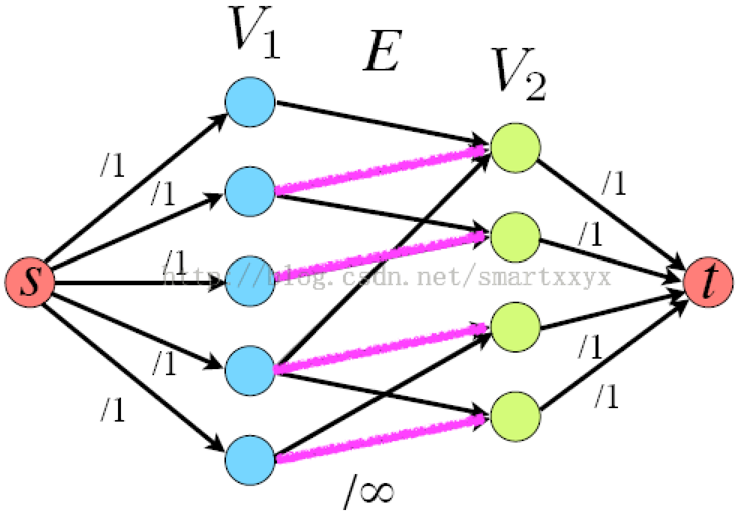




最大匹配：一个图所有匹配中，所含匹配边数最多的匹配，称为这个图的最大匹配。图 4 是一个最大匹配，它包含 4 条匹配边。

二部图的最大匹配

对于一个二分图，令已有的边的容量为无穷大，增加一个源点s和一个汇点t，令s和t分别连接二部图中的一个分部，并设置其容量为1。这时得到流网络G'，计算得到的最大流就等于最大二分匹配。



二分图匹配例题

POJ

1274， 2239， 2584(二分图多重匹配)， 2536， 2446