网络流_11 最小割之最大权闭合图、最大密度子图

最小割之最大权闭合图

简单割定义：

所有割边都与源点或者汇点相连,由

$|f|_{最大流} = max\{\sum_{u\in V} f(s,u)\}$=最小化割 <= 所有割，因此割边容量一定有限，因此此情况下最小割一定是简单割。

闭合图定义：

定义有向图$G=(V,E)$为一闭合子图当： $\forall v \in V$ 对应出边的端点v'都有$v' \in V$。简而言之，所有选的点集和边集中所有点集内点的出边都在边集中即可。

最大权闭合图：

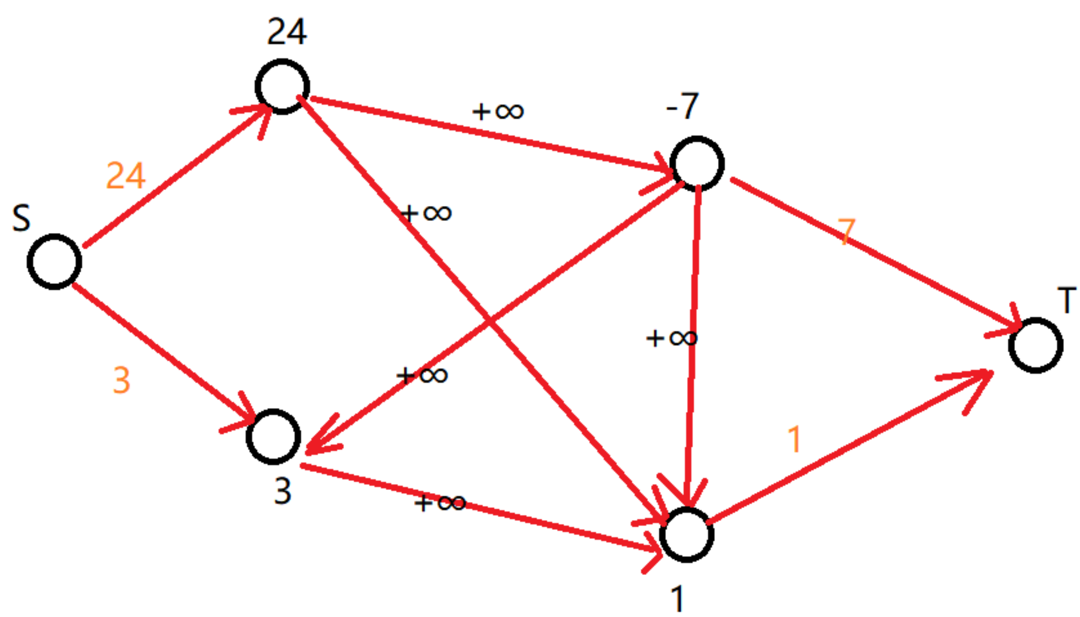
即再给出的边集以及边的关系中求得点权之和最大的闭合图集合

条件：

G为有向图。

一般性建图：

> 一堆点分别由正权点和负权点构成，每个点都至少有一种指向关系。建立源点和汇点，源点向所有正权点连对应权值的容量，所有负权点向汇点连对应绝对值权值的容量，每个点之间的转移容量为正无穷。此图中的最小割=最大流即为**最大收益**(描述的是一种正权减负权的关系)



证明 最小割情形下的闭合子图为最大权闭合子图：

由于不存在`S->T`的边，又因为是简单割，因此不存在内部点跨越两个割`v1->v2`,因此只存在`S->v2`和`v1->T`的两种边。其中V_1指的是选出的闭合图，V_2指的是选出的闭合图的补集

$C[S,T]=C[V_1,\{t\}]+C[\{t\},V_2]=\sum_{v\in V_2^+} W_v + \sum_{v\in V_1^-} -W_v$ (根据是简单边而得来)

（最优性）又有$W(V_1)=\sum_{v\in V_1^+}W_v-\sum_{v\in V_1^-}(-W_v)$

$C[S,T]+W(V_1)=\sum_{v\in v_2}W_v$,其中$\sum_{v\in v_2}W_v$为正权点集和，为定值，因此为了最大化$W(V_1)$,因此当为最小割的时候成立

例题：最大获利

> n个设施，建设耗费P_i花费，m个用户，第i个用户需要依赖一些设施才能使得你盈利C_i。问你如何建设才能获得最大盈利，求盈利

分析：因为每个点具有点权P_i或者C_i，且部分点有依赖，我们将其抽象为一个闭合子图，仿照闭合子图的建图方法，$S->^{C_i}$ 用户, 设施$->^{P_i}T$，由最大闭合子图的公式
$Sum_+ - C[S,T]=f_{最大权闭合子图}$求解

```
void solve(){
    n=read(),m=read();
    S=0,T=n+m+1;
    rep(i,1,n){
        int x=read();
        add(i,T,x);
    }
    rep(i,1,m){
        int a=read(),b=read(),c=read();
        add(S,i+n,c);
        add(i+n,a,INF), add(i+n,b,INF);
        tot += c;
    }
    print(tot-dinic());
}
```

例题 Magic Slab

> 一个方格图，数组`a[i]`,`b[j]`分别表示选择行`i`和列`j`的代价。每个格子都有个权值，当且仅当行和列的代价同时选上时会获得当前各自内权值的收益。此外有附加提交，即同时选上指定的两个格子时能获得一个额外的`e[i]`的收益；

建图方式与最大权闭合子图完全相同，额外收益的转化方式为新增加一个收益点连向四个代价点即可。

最大权闭合子图的方案

从源点开始沿着大于0的正向边搜索，能够搜到的点都是左集合，即选择的方案中的点。右边所有被割掉的点都是选择对应的必选项。

最大密度子图

定义：

定义一个无向图$G=(V,E)$,其中所有边的两个端点均被选择。即可以选择所有点但是不选边，但是选了边对应的点也要选

定义无向图$G=(V,E)$的密度D为该图的边数$|E|$与该图的点数$|V|$的比值$D=\frac{|E|}{|V|}$。给出一个无向图$G=(V,E)$,其密度最大的子图称为最大密度子图，及要求最大化D

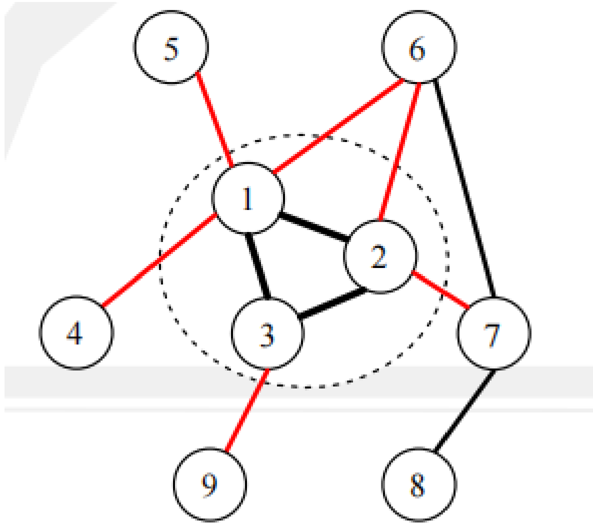
求解方法： `01分数规划+最小割定理`

$\frac{|E|}{|V|}=g$ 二分答案g，每次最大化 $|E|-g*|V|$，即最小化$g*|V|-|E|$(因为最小割定理解决的是最小化问题)，如果小于0，向左缩小区间，否则向右缩小区间

引理：

对于图G的子图$G^{'}=(V^{'},E^{'})$在点集固定的情况下必然选的边越多越好，因此把选出点集所形成的所有边都选上就得到了一个G的导出子图，此方案下比其他方案更优

然后使用逆向思维：用所有边减去与$V^{'}$关联的且不符合条件的边就可以得到$E^{'}$，删去的边如红色所示：



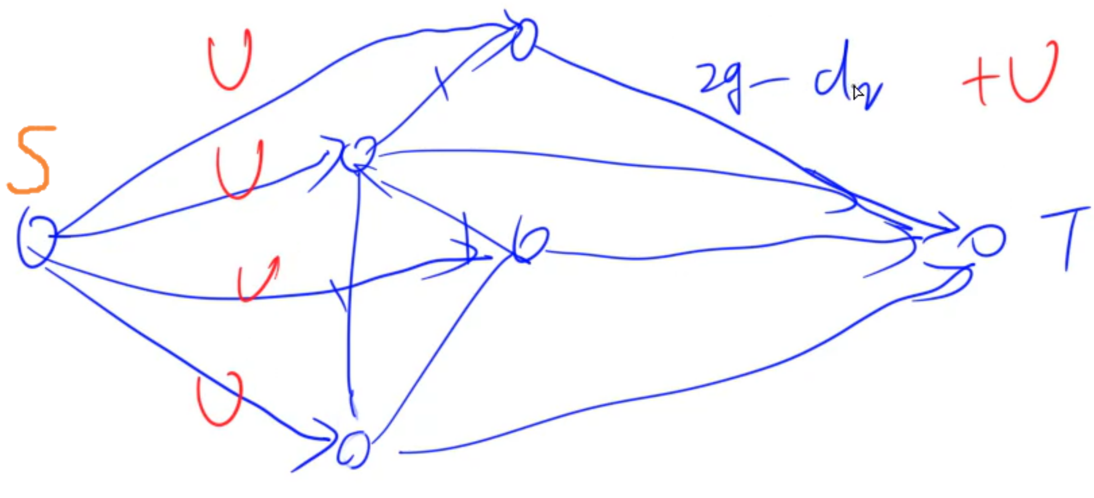
论文《最小割模型在信息学奥赛中的应用》提出红色的边实际上就是`v'`与其补集合$\overline{V'}$之间的边，因此尝试使用最小割定理。继续化简式子：

$g*|V|-|E|$

$=\sum_{v\in V'}g-(\frac{\sum_{v\in V'}deg_v}{2}-\frac{c[v',v'的补集]}{2})=\sum_{v\in V'}(g-deg_v/2)+c[V',V'的补集]$

$=\frac{1}{2}\cdot(\sum_{v\in V'}(2g-d_v)+c[V',V'的补集])$

我们发现对于$\sum_{v\in V'}(2g-d_v)$处理方式是相当于选了这个点就要花费`2*g-deg`的代价，相当于负权边，因此建图方式为从此点向汇点连一条此绝对值容量的边，但是还有一个问题就是`2*g-d`有可能是是负数，因此处理方式就是加上一个偏移量`U`使得其为正数。因此最终的建图方式就是(中间的边权为1)：



此时$c[S,T]=\sum_{v\in V'}U+\sum_{v\in V'}(U+2\cdot g-d)+\sum_{(u,v)\in E}1=U\cdot n-2\cdot(|E'|-g|V'|)$

生活的艰辛

> 在一个图`G`中找到一个最大密度子图

通过如上建图后跑最大流不断二分答案即可。

拓展1 有边权

如果有边权`we`的话，则对密度定义进行扩展为$D = \frac{\sum_{e \in E} W_e}{|V|}$

由于每条边赋上边权，则目标是最小化$g \cdot |V'| - \sum_{e \in E'} w_e$，定义$d_u = \sum$与$u$点相连的边

因此新的建图方式为:

$$c(u,v) = c(v,u) = w_e \qquad c(S,v) = U \qquad c(v,T) = U + 2g - d_v$$

拓展2 同时有边权和点权

此时最大化的密度D定义为$D = \frac{|W_e'| + |P_v'|}{|V'|}$,根据二分答案，有:

最小化$\sum g - \sum |P_v'| - \sum |W_e'| = \sum (g - |P_v'|) - \sum |W_e'|$，同样定义$d_u = \sum$与$u$点相连的边

因此新的建图方式为:

$$c(u,v) = c(v,u) = w_e \qquad c(S,v) = U \qquad c(v,T) = U + 2g - d_v - 2p_v$$