深 圳 大 学 实 验 报 告

课程名称：算法设计与分析

实验项目名称：最大流应用问题

学院：计算机与软件学院

专业：计算机科学与技术

指导教师：杜智华

报告人：欧阳宇杰 学号：2021150143 班级：计科 2 班

实验时间：2023 年 6 月 4 日~ 6 月 15 日

实验报告提交时间：2023 年 6 月 15 日

教务部制

一、实验目的

（1）掌握最大流算法思想。

（2）学会用最大流算法求解应用问题。

二、实验内容 论文评审问题

1. 有 m 篇论文和 n 个评审，每篇论文需要安排 a 个评审，每个评审最多评 b 篇论文。请设计一个论文分配方案。

2. 要求应用最大流解决上述问题，画出 m=10，n=3 的流网络图并解释说明流网络图与论文评审问题的关系。

3. 编程实现所设计算法，计算 a 和 b 取不同值情况下的分配方案，如果没有可行方案则输出无解。

三、实验要求

1. 在 blackboard 提交电子版实验报告，注意实验报告的书写，整体排版。

2. 实验报告的实验步骤部分需详细给出算法思想与实现代码之间的关系解释，不可直接粘贴代码（直接粘贴代码者视为该部分内容缺失）。

3. 实验报告中要求证明该算法的关键定理，并说明这些定理所起的作用。

3. 实验报告样式可从 http://192.168.2.3/guide.aspx 表格下载－学生适用－在校生管理－实践教学－实验：深圳大学学生实验报告）

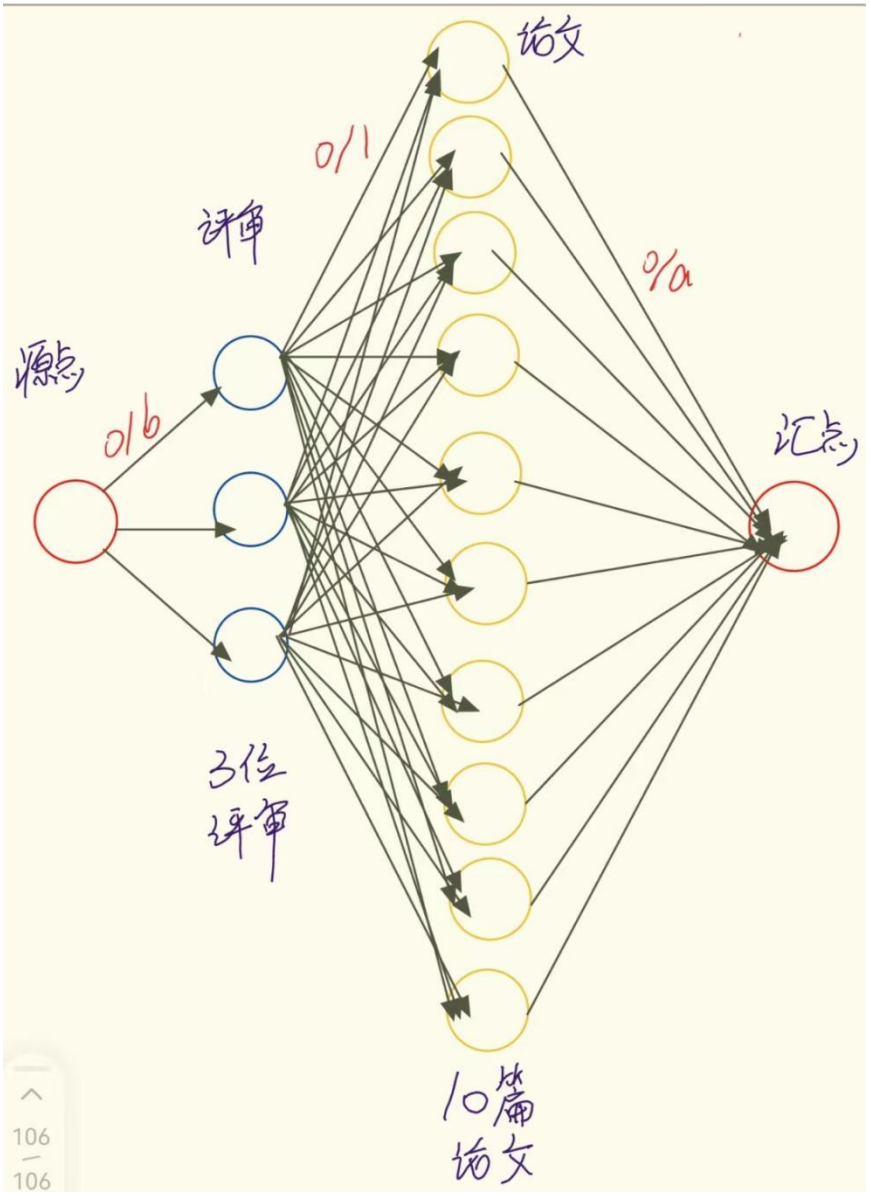
4. 源代码作为实验报告附件上传。

5. 在实验课需要现场运行验证并讲解 PPT。

四、实验过程与步骤

❖ 构建流网络图

因为要利用网络流解决问题，所以首先要构建流网络。为了让流网络结构更加清晰，我分层地构建流网络。下图为 m=10,n=3（10 篇论文，3 位评审）时构建的流网络图。



根据题意，可以构建出如上的流网络。流网络分为 4 层。第一层为源点，第四层为汇点。第二层表示各个评审，第三层表示待评审的论文。

源点→评审：源点分别指向各个评审节点且容量为每个评审的最大评审论文数。由于需要对各个论文进行评审，所以源点与每个评审间存在有向边。又因为每个评审最多可以评审 b 篇论文，因此，从源点到评审节点的最大流量为 b。

评审→论文：对于每个评审，评审需要评阅论文，因此存在由评审到论文的有向边。又因为对于每个评审，其评阅论文的状态只有 0 或 1 即表示未评阅或已评阅。因此从评审到论文的有向边的最大流量为 1。

论文→汇点：完成论文的评阅后将流网络汇入汇点即可。又因为，每篇论文需要 a 个评审进行评阅，则从论文节点到汇点有最大容量为 a 的有向边。当论文 w 到汇点的流量为 x 时，表示论文 w 被 x 个评审评阅了。

❖ 流网络中最大流与论文评审问题解的关系

流网络最大流下各边的流量就是一组值班问题的解：

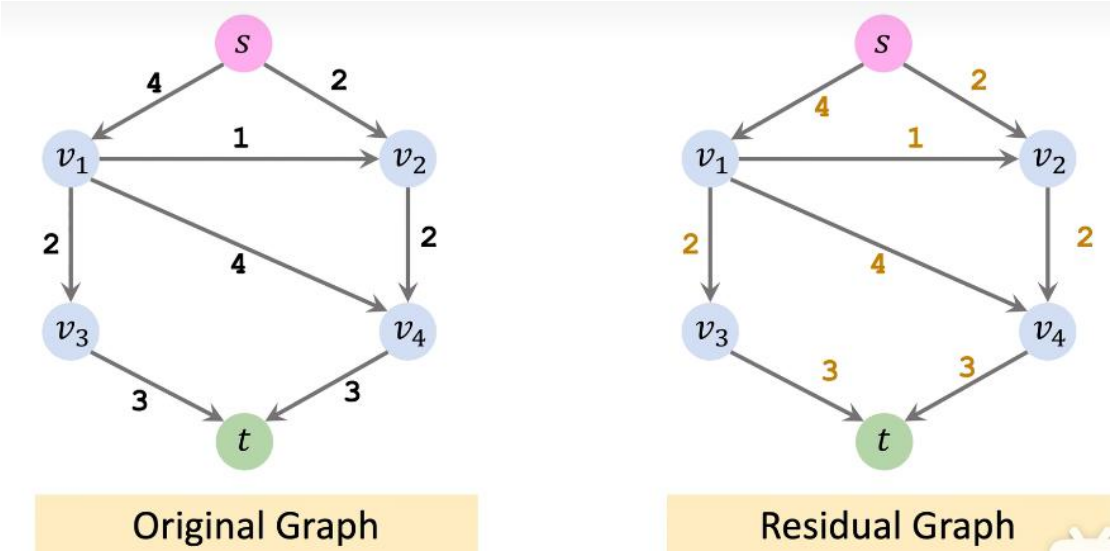
当流网络流入汇点总量不为论文数与每篇论文需要评审数的积（m×a）时，说明有部分论文没有被规定数量（a）的评审评阅，此次论文评审问题无解；

显然如果这个问题有解，即每篇论文都有 a 位评审进行评阅，那么从论文层到汇点间边的流量必为 a。此时，论文层到汇点间所有边的容量都全部流满，则此时流网络必为最大流，即论文评审问题的解一定对应流网络中的最大流情况。论文评审问题是否有解的依据就是最大流是否为论文数与每篇论文需要评审数的积（a*m）。

❖ 最大流的计算

Edmonds-Karp 算法流程：

首先建立一个残差图；残差图上边的数字表示表示边的空闲量，初始化为容量。



不断循环：如果能找到增广路径就继续以下操作

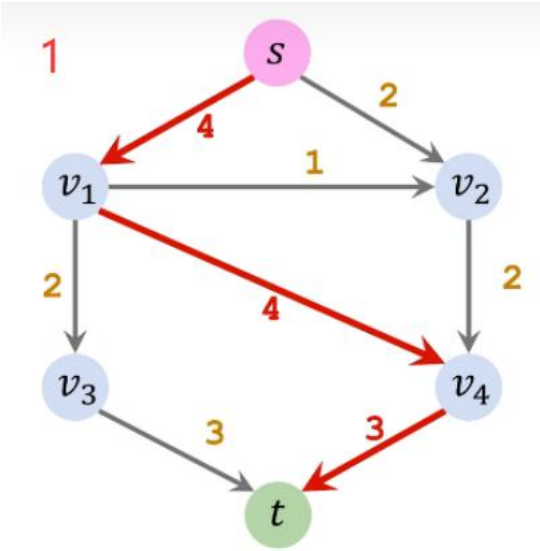
1.找到最短的增广路径 (在残差图上使用 BFS.并且在增广路径上找到瓶颈量 x）

2.更新残留图(增广路径上的边空闲量-刚刚找到的瓶颈量 x)

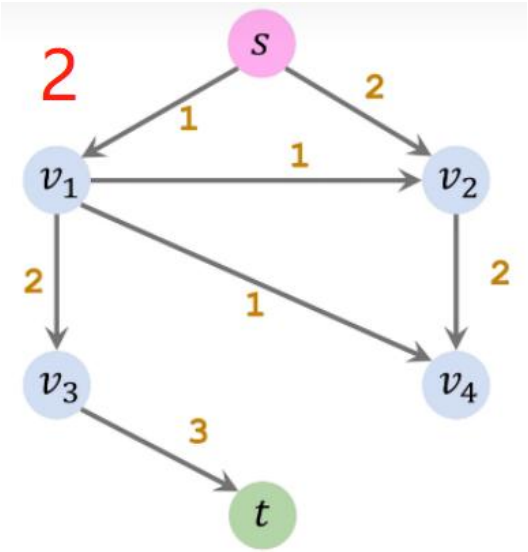
3.在残留图上增加反向边(沿着刚刚的增广路径，每条反向边的容量都为 x)

以下面的例子展示 Edmonds-Karp 算法

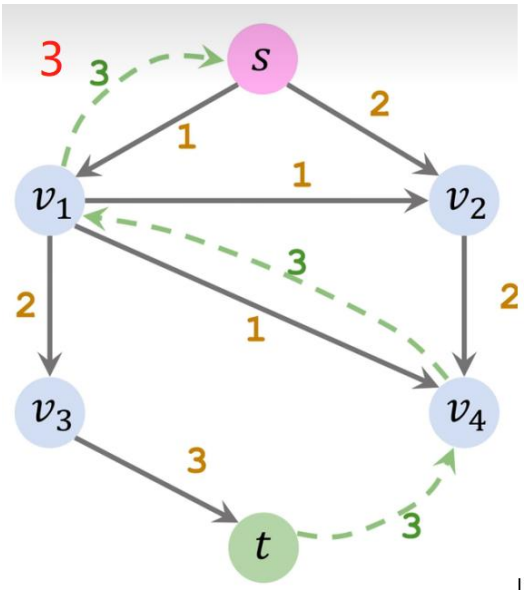
1.找到最短的增广路径 (在残差图上使用 BFS.并且在增广路径上找到瓶颈量 x）



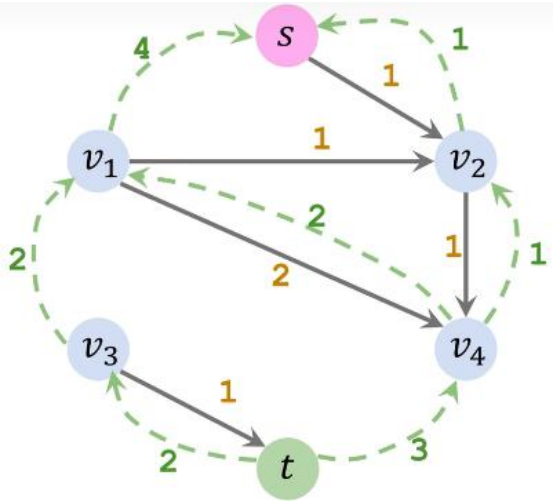
2.更新残留图(增广路径上的边空闲量-刚刚找到的瓶颈量 x)



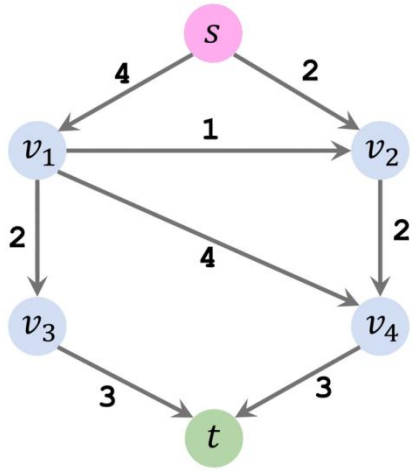
3.在残留图上增加反向边(沿着刚刚的增广路径，每条反向边的容量都为 x)



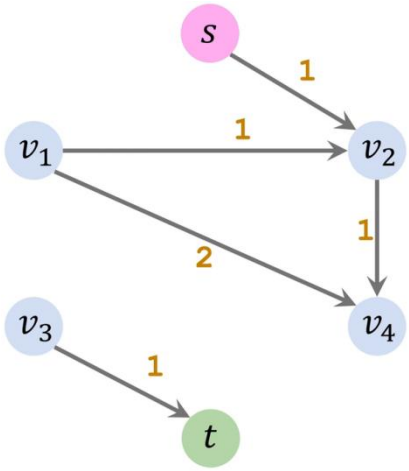
不断地进行如上的循环，直到找不到增广路径就退出循环。退出循环时的结果为：



此时不再需要反向边，去掉反向边得到下面的右图，

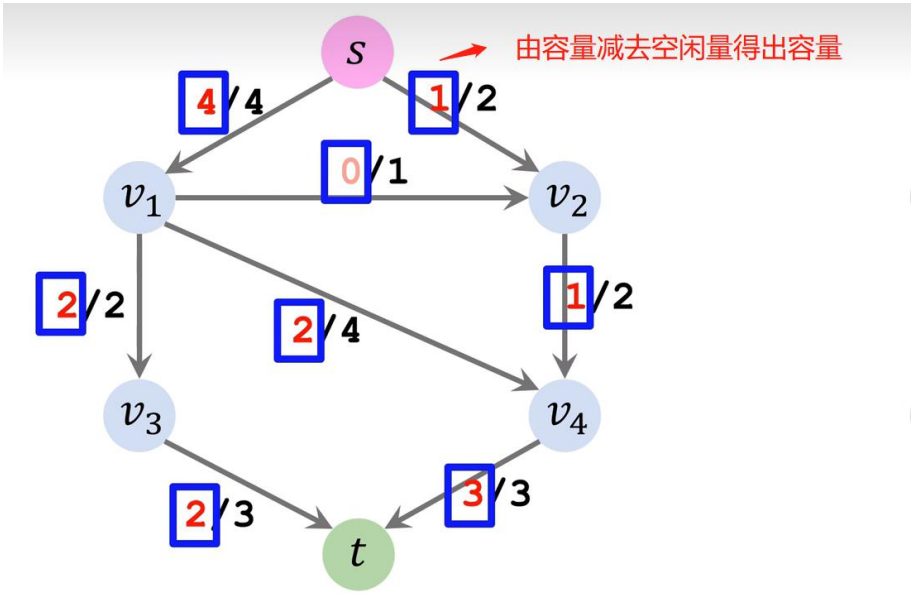


Original Graph



Residual Graph

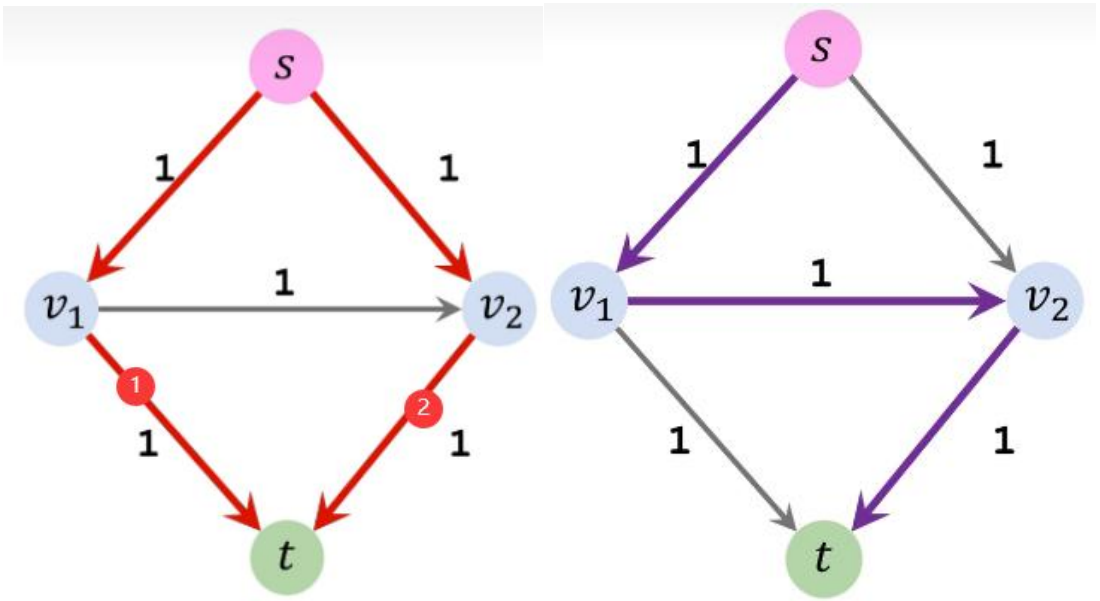
左右两张图中权重之差就是管道中的流量，得到下图：



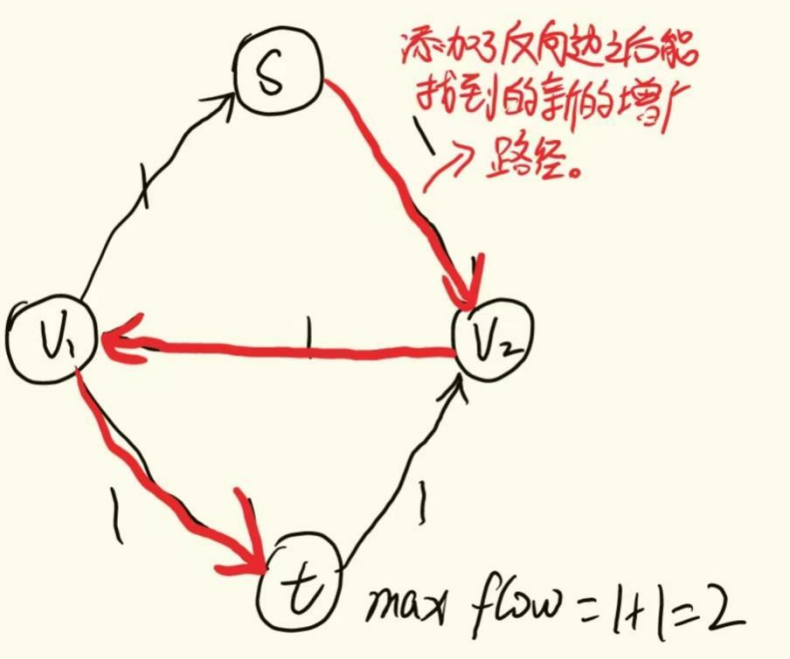
则由上图可知该图的最大流为 5。

为什么要添加反向边？

增广路搜索的顺序不同可能会导致汇入汇点的流量不同。导致最大流求解错误。如下图，如果寻找的两条增广路径为下面左图的两条，则能得到正确的最大流 2。但若先找到下面右图中蓝色的路径，则会导致流入汇点的流量为 1，且无法继续增加，导致最大流求解错误。



但若找到了增广路径后添加了反向边，那么则就能再找到一条增广路径，流入汇点的流量就能再加 1，变为 2，得到正确的最大流。

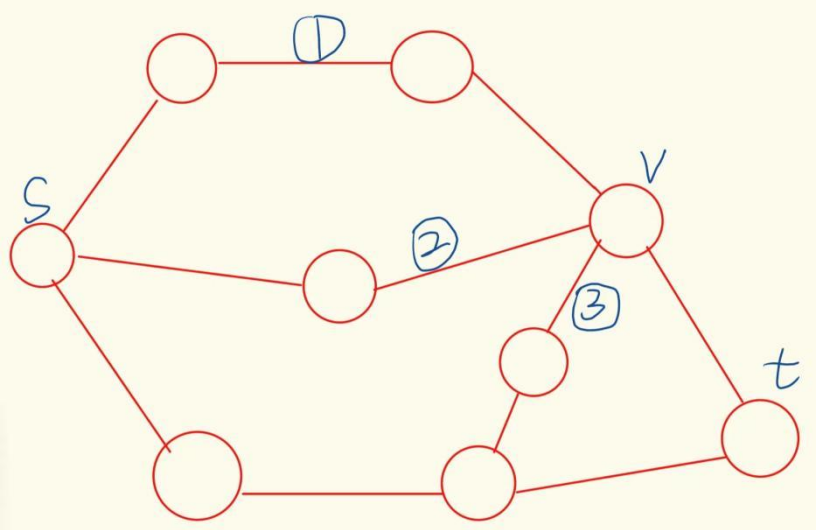


添加了反向边之后能够保证不管增广路径的搜索顺序如何都能得到正确的最大流结果。

❖ 算法关键定理证明：

定义：min_dis[c]表示 c 点到源点 s 点的最短距离。

引理：如果 Edmonds-Karp 算法运行在流网络 $G = (V, E)$上，该网络的源结点为 s，汇点为 t，则对于所有的结点 $v \in V - \{s, t\}$，残存网络G_f中 min_dis[v]随着每次流量递增而单调递增。



引理的证明：流量递增则对应着一次增广。如上图中的 v 点，易得此时 min_dis[v]为 2，对应着②号路径。如果增广导致最短路径被破坏，s 点无法通过原来的最短路径到达 v 点，那么 s 点就只能通过更长的路径如到达 v 点，那么 min_dis[v]就会增加。如果增广导致 s 到 v 的非最短路径被破坏或者增广并未导致 s 到 v 的任何路径被破坏，那么自然对 min_dis[v]没有影响，因为 s 仍然可以沿着最短路径到达 v。故 min_dis[v]随着不断地增广而单调递增。

定理：如果 Edmonds-Karp 算法运行在源结点为 s，汇点为 t 的流网络 G=(V，E)上，则该算法所执行的流量递增操作的总次数为 O(VE）。

证明：若增广路 p 的残留容量等于边(u,v)的残留容量，则称边(u,v)是增广路 p 的关键边，下面用引理证明每条边最多做关键边|V|/2 次。

由于增广路径都是最短路径，所以边(u,v)第一次成为关键边时有：

$$min\_dis[v] = min\_dis[u] + 1$$

而增广之后，(u,v)将会从残留网络中消失，重新出现的条件是(v,u)出现在增广路上。因为若(v,u)出现在增广路上，那么(u,v)就会被添加进残留网络。那么则有：

$$min\_dis'[u] = min\_dis'[v] + 1$$

由引理我们知道：　$min\_dis'[v] >= min\_dis[v]$

故有 $min\_dis'[u] = min\_dis'[v] + 1 >= min\_dis[v] + 1 = min\_dis[u] + 2$

因此，边(u, v)重新出现一次，从源节点 s 到节点 u 的最小距离至少增加 2 个单位，而从源节点 s 到节点 u 的最初距离至少为 0，从 s 到 u 的最短路径上的中间结点不可能包括 t（因为边(u, v)处于一条增广路径上意味着 u≠t），因此 s 到 u 的最短路径至多含有$|V| - 1$ 个结点，因此 s 到 u 的最短路径最长为$|V| - 2$。因此边(u, v)至多重新出现$|V|/2 - 1$ 次。由于出现的前提是消失，也就是成为关键边，则边(u, v)至多成为关键边$|V|/2$ 次。

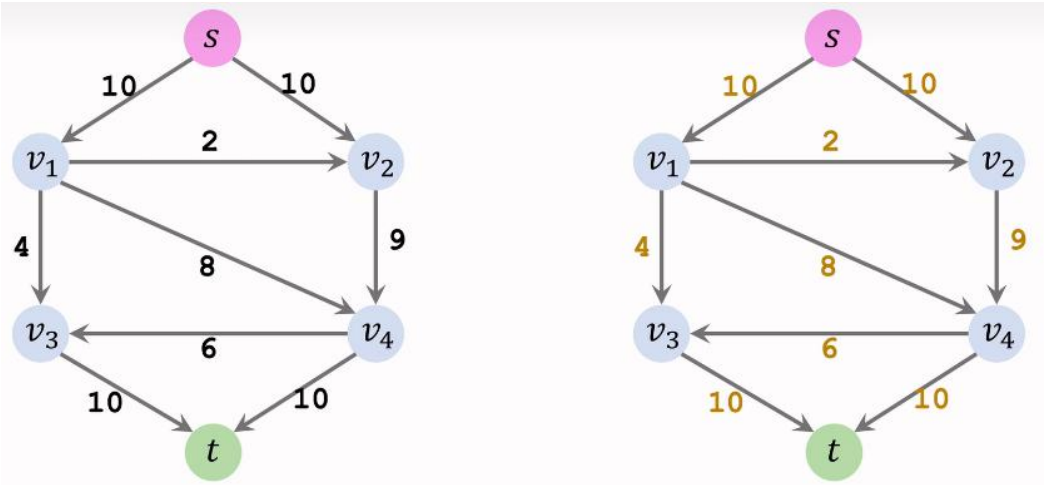
通过前面我们已经知道了一条边最多成为关键边$|V|/2$ 次，又知道共有 E 条边，那么关键边最多为$|V|E/2$ 个。每次循环都会找到一条增广路径，每条增广路径至少有一条关键边，那么表明循环至多进行$|V|E/2$ 次，则证明了定理：算法所执行的流量递增操作的总次数为 O(VE)。

❖ EK 算法时间内复杂度

在前面的定理证明环节已经知道了循环至多进行 O(VE)次。又因为每次循环的 BFS 需要时间 O(E)，那么总的时间复杂度为 $O(VE^2)$

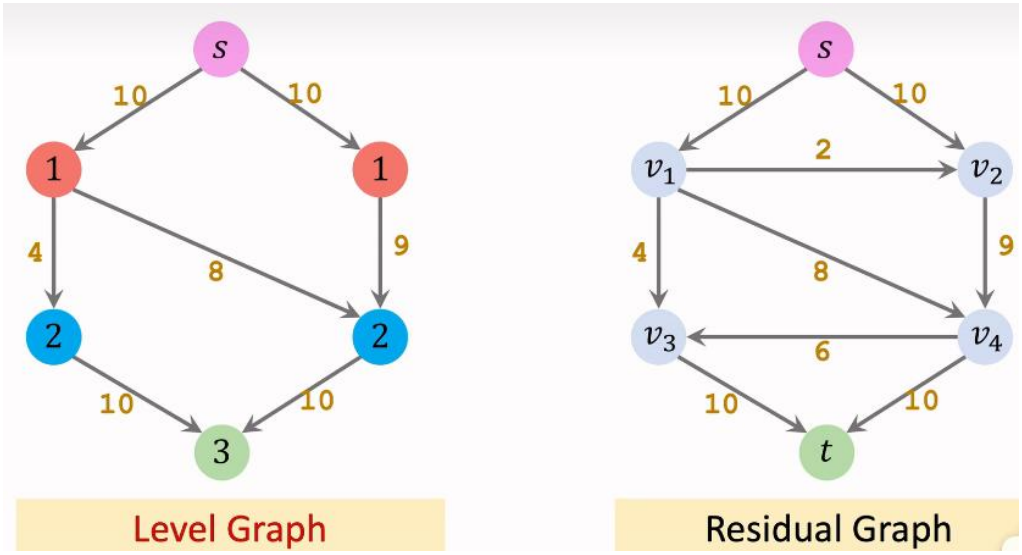
❖ Dinic 算法流程：

初始化残留图（下面右图）和原图（下面左图）一样：



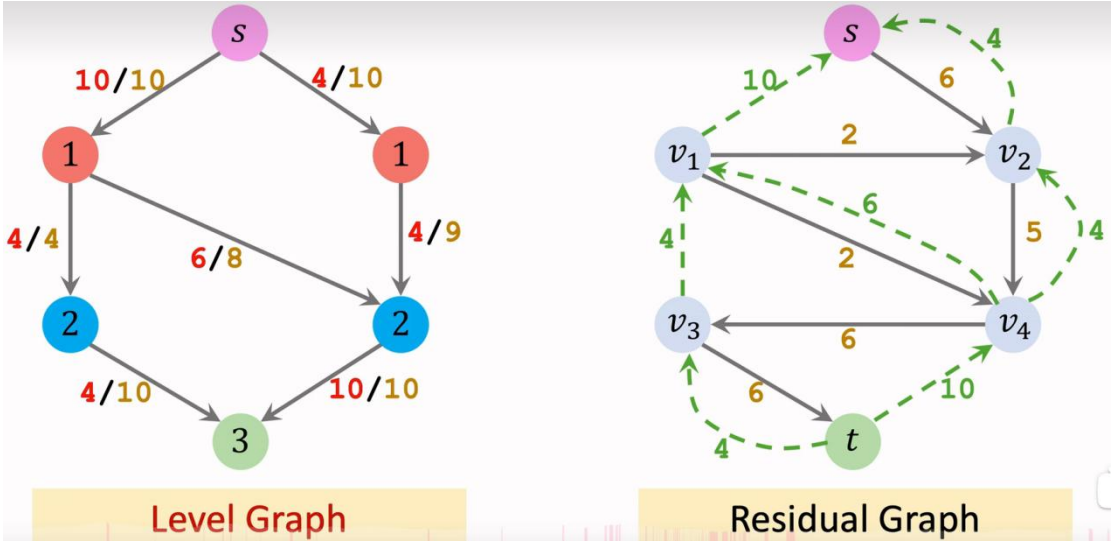
循环下面直到退出：

1. 根据残留网络建立层次图，如下（左图为层次图，右图是残留网络。在层次图中非低层次到高层次的边之后的 DFS 不可能走，故未画出）。



2. 在当前层次图上不断地用 DFS 寻找增广路径直到再也不能找到增广路径。寻找到一条增广路径后就要更新残留图（更新增广路径上边的权重，并加入反向边）并增加流入汇点的流。但由于反向边是高层次往低层次的边，所以 DFS 无法利用反向边。所以这步的不断 DFS 不能保证最大流。

下面左图是再也找不到增广路径时的层次图，右图是此时对应的残留图。



3. 回到第 1 步，根据残留图建立层次图，直到无法建立层次图算法完成。

算法实现伪代码：

```
bool dinic()
    result=0,flow;
    while (bfs())
        while (flow = dfs(s,INF))
            result += flow;
    return result;
```

```
bfs()
    queue Q;
    Q.push(s)
    for(i = s to t)
        level[i]=0;
    level[s]=1;
    while(!Q.empty())
        i = Q.front();
        Q.pop();
        for(j = s to t)
            if(!level[j]&&edge[i][j]>0)
                level[j]=level[i]+1;
                Q.push(j);
    return level[t]!=0
```

```
dfs(u,p)
    temp=p;
    if(u == t)
        return p;
    for(j=s;j<=t&&temp;j++)
        if(level[j]==level[u]+1&&edge[u][j]>0)
            t = dfs(j,min(temp,edge[u][j]));
            edge[u][j] -=t;
            edge[j][u] +=t;
            temp-=t;
    return p-temp;
```

❖ 算法时间复杂度：

Dinic 算法每一次建立层次图，需要执行一次 BFS 操作，时间复杂度为 O(E)。建立完层次图后，我们要不断地用 DFS 寻找增广路径，由于一条增广路径对应着至少一条关键边，关键边的个数为 O(E)，所以寻找增广路径的次数为 O(E)次。又因为在层次图中用 DFS 找增广路径，DFS 时只能从低层次的点到高层次的点，所以 DFS 的时间复杂度为 O(V)，那么一次分层导致的时间复杂度为：

$$O(V + E) + O(VE)$$

而最多会建立 O(V)次层次图，所以 Dinic 算法的时间复杂度为 $O(V^2E)$。

❖ 下面根据实验要求计算 a 和 b 取不同值情况下的分配方案

在前面已经给出：论文评审问题是否有解的依据就是最大流是否为论文数与每篇论文需要评审数的积(a*m)。m 为 10，n 为 3。

EK 算法结果：

a	b	最大流	时间消耗	是否有解
1	20	10	0.382ms	有解
2	16	20	0.704ms	有解
3	12	30	0.949ms	有解
4	5	15	0.396ms	无解
3	2	6	0.173ms	无解

Dinic 算法结果：

a	b	最大流	时间消耗	是否有解
1	20	10	0.033ms	有解
2	16	20	0.031ms	有解
3	12	30	0.024ms	有解
4	5	15	0.019ms	无解
3	2	6	0.018ms	无解

❖ 对比 Dinic 算法和 EK 算法

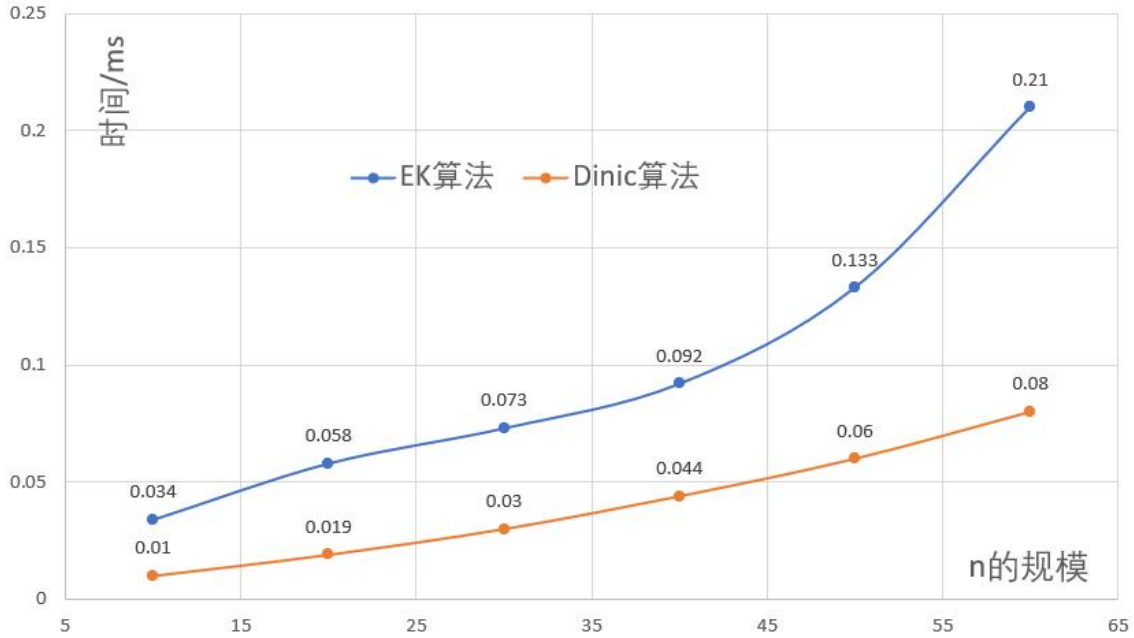
针对实验给出的问题，我指定 b=13，a=3，m=10 不变，改变 n（评审数量）的大小，对比 EK 算法和 Dinic 算法的运行时间。

可以得到以下的运行时间对比表格：

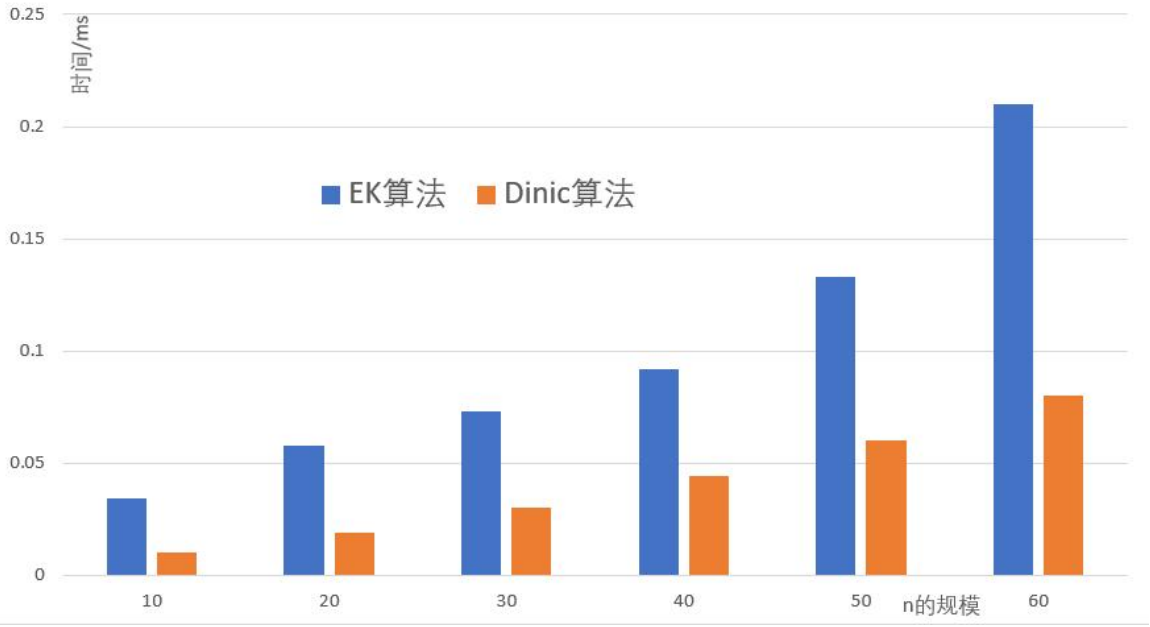
n的大小	10	20	30	40	50	60
EK算法	0.034	0.058	0.073	0.092	0.133	0.21
Dinic算法	0.01	0.019	0.03	0.044	0.06	0.08

根据上面的表格可得出下面的运行时间对比曲线图和对比条形图。

运行时间对比曲线图：



运行时间对比条形图：



通过上面的表格和图象可以看到 Dinic 算法的实际表现更为优秀，算法运行时间少于 EK 算法。

五、实验总结与体会

生活中的一些排班问题、分配问题可以通过网络流的方式进行解决。这时求解最大流问题的算法就能够起作用了。

通过此次实验，我认识并理解了求解最大流问题的 Edmonds-Karp 算法和 Dinic 算法，这两个算法都能够求解最大流问题，Edmonds-Karp 算法的时间复杂度为 O（VE^2），Dinic 算法的时间复杂度为 O（V^2E）。由于在实际的图中边的个数 E 往往大于点的个数 V，所以 Dinic 算法的实际表现更为优秀，算法的运行时间往往优于 Edmonds-Karp 算法。

通过本次实验我对于最大流应用、BFS 以及 DFS 的理解更加深入了。我很高兴能够通过此次实验习得网络流与最大流的相关知识。这增加了我的算法知识储备。

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。