**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 算法设计与分析**  **项目名称 最大流应用问题**  **学 院 计算机与软件学院**  **专 业 软件工程**  **指导教师 卢亚辉**  **报 告 人 郑彦薇 学号 2020151022**  **实验时间 2022/6/7~2022/6/21**  **提交时间 2022/6/19** |

**教务处制**

# 一、实验目的与要求

**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、实验内容与方法

**（一） 实验内容**

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。

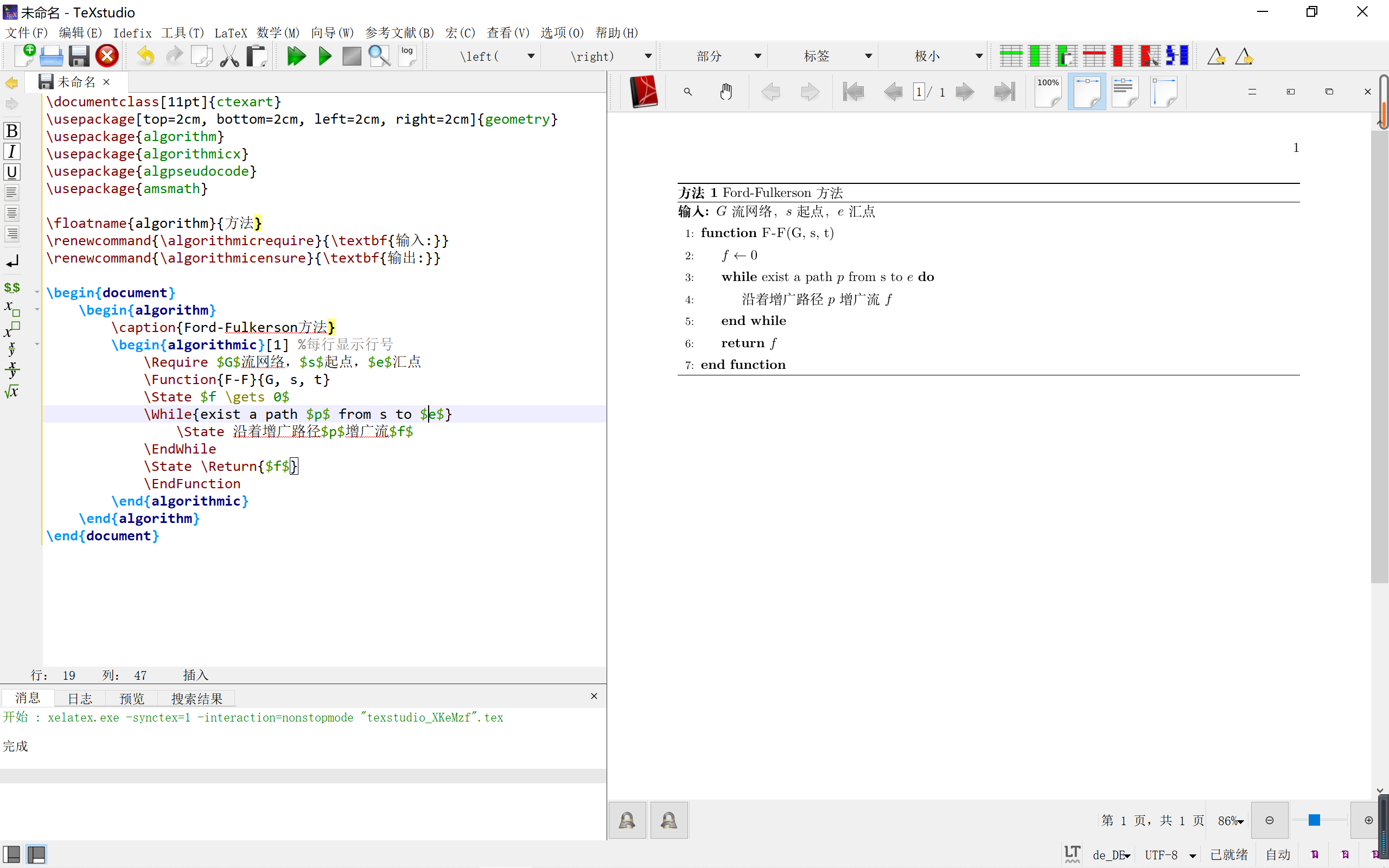
3. 编程实现所设计算法，计算a和b取不同值情况下的分配方案，如果没有可行方案则输出无解。

**（二）求解方法**

根据实验内容知道，我们需要应用最大流解决上述论文--评审分配问题，而求解最大流问题的方法有Ford-Fulkerson方法。

1. **方法和相关概念引入**
2. Ford-Fulkerson方法（以下简称FF方法）：FF方法是一种求解最大流问题的迭代方法，它依赖于三种重要思想：残留网络、增广路径和割。

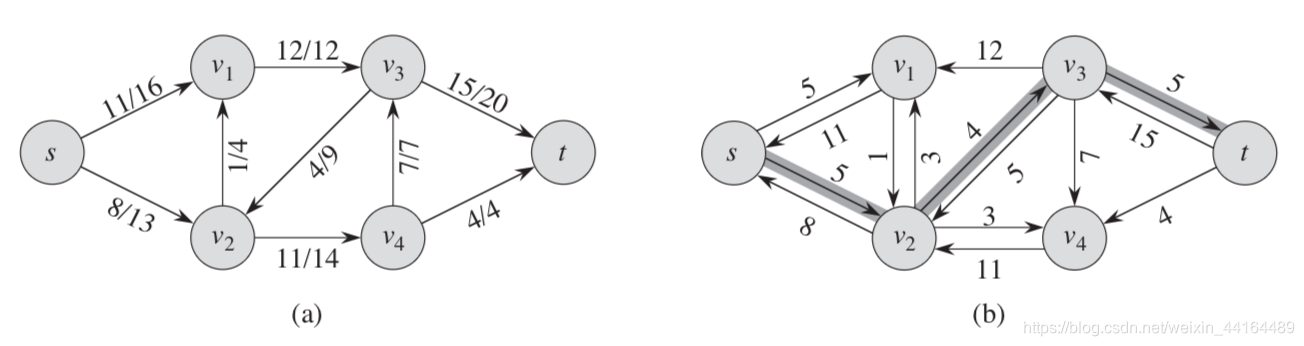
方法基本伪代码：



1. 残留网络：残留网络就是有残留容量的图，即在图中增加一条反向边，反向边的值为该边的流的大小，原始边的值为边的容量减去边的流大小。如下公式所示：



通过下图进行进一步说明，左图所示的流网络对应的残留网络如右图所示：



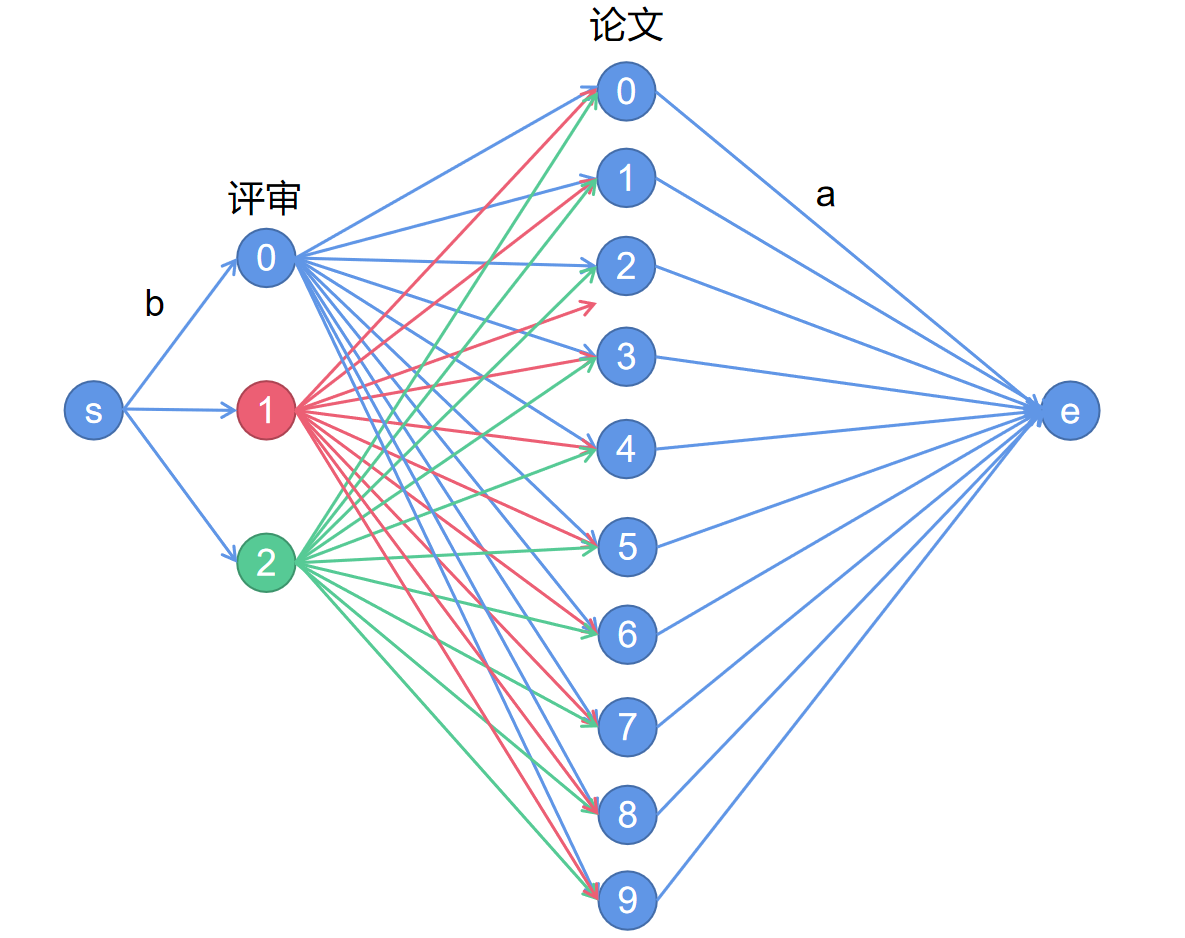
1. 增广路径：增广路径是指在一个残留网络中，一条从源点s到汇点e的简单路径。当残留网络确定时，我们可以通过BFS或DFS求得增广路径，这也是问题求解的关键。
2. 最大流最小切割定理：该方法告诉我们，一个流是最大流当且仅当它的残留网络中不包含增广路径。这个定理是求解问题的基础。

# 三、实验步骤与过程

**（一）问题转换**

根据上面所提相关概念和方法，我们可以知道解决论文和评审的分配问题，可以将该问题转化为最大流问题。

已知对于m篇论文，n位评审，规定一篇论文需要安排a个评审，一个评审最多可以评b篇论文。将论文和评审转换为节点，将评审可以处理的论文数b作为评审的入度，将每个论文需要安排的评审数作为论文的出度，加上首尾的起点s和汇点e，构造该问题的流网络如下图所示：

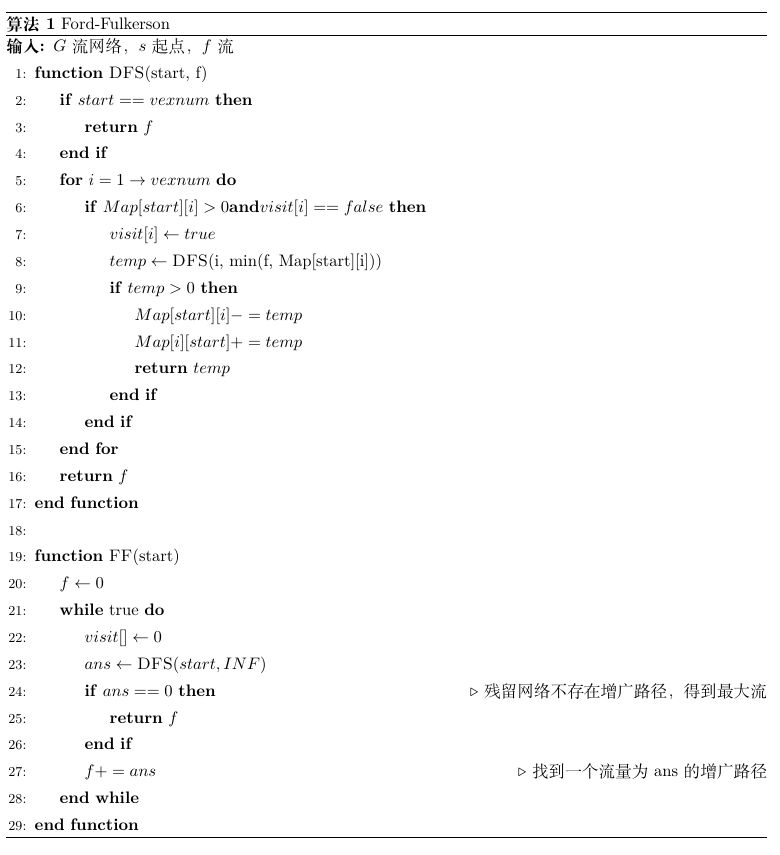


对于a个评审，总共需要处理的论文数为a·m篇，因此判断是否存在可行的分配方案则是判断所求的最大流等不等于am，若等于，说明存在解决方案；若不等于，说明不存在可行的分配方案。

综上，我们可以得到解决问题的整体思路为：得到流的残留网络→使用DFS或BFS获得残留网络中的增广路径→根据增广路径是否存在判断流是否是最大流→若这个流为最大流（增广路径不存在），判断最大流与am是否相同→得到结果。

1. **FF基本法**
2. 解题思路
3. 根据上述问题转换，首先根据输入存储图信息，包括源点到评审、评审到论文、论文到汇点的边信息以及权值。其中源点到评审的权值为评审最多可以评审的论文数b，评审到论文的权值为1，论文到汇点的权值为评一篇论文需要的评审数a。
4. 使用DFS寻找从源点到汇点的增广路径，当残留网络中还存在增广路径时，记录下当前DFS返回的流量值，继续通过DFS查找是否还存在增广路径。若增广路径不存在（即DFS返回值为0），则返回当前总流量值即为最大流。
5. 记录通过上述DFS递归查找得到的最大流的值，与论文数和论文所需评审数的乘积（即a\*m）进行比较，如果该值可以等于am（即am<=最大流），则存在分配方案；否则不存在。
6. 伪代码

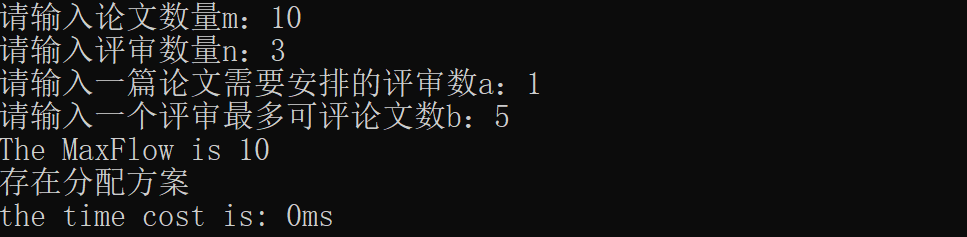
根据思路，编写伪代码如下：



1. 算法正确性验证

对于10篇论文、3个评委，将a赋值为1，b赋值为5。假设评审编号为A、B、C，论文编号从0~9，可以任意得到一种分配方案为：A负责0~4，B负责5~9.（存在多种方案，这里只验证是否有分配方案的判断的正确性）。

输入相应的值，得到结果如下：



算法正确性得以验证。

1. 性能分析

固定论文数为500，评审数为100。

1. 固定a的值为50，增加b的值统计运行时间：



1. 固定b的值为30，增加a的值统计运行时间：



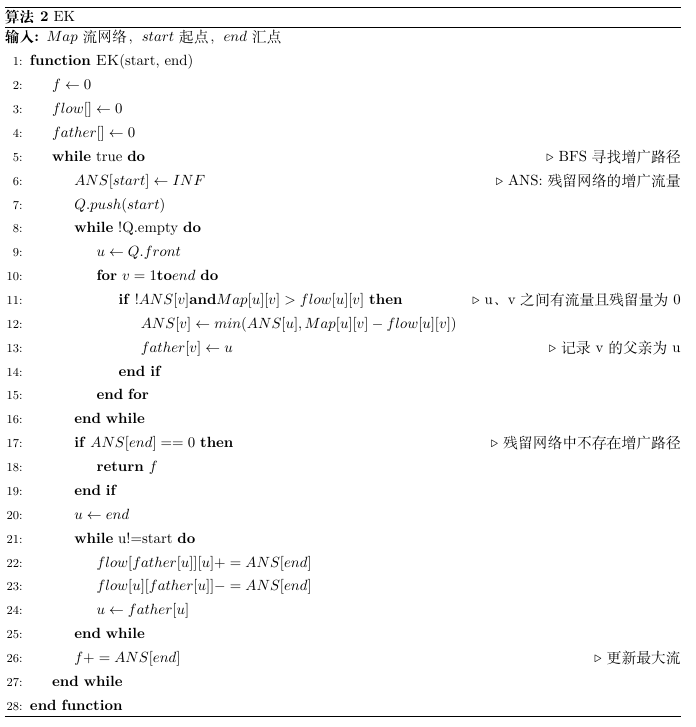
通过上述对运行时间的统计以及FF算法的最坏时间复杂度可以知道，算法实际运行时间与最大流有关，运行时间随最大流值的增大而增大；而最大流保持不变时，运行时间又会随a的增大而减小。

1. **Edmons-Karp算法（以下简称EK算法）**
2. 解题思路

该算法的核心思想与FF算法一致，只是使用BFS代替DFS进行增广路径的查找。

1. 伪代码

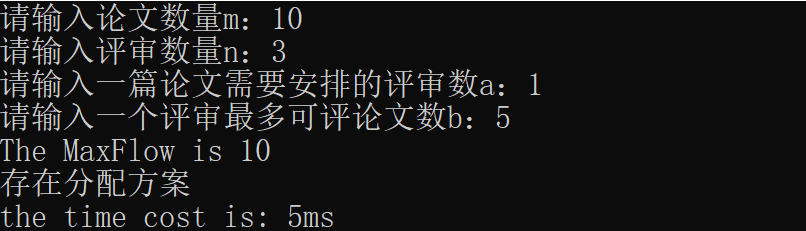
根据思路，编写每一个模块功能的伪代码如下：



1. 算法正确性验证

对于10篇论文、3个评委，将a赋值为1，b赋值为5。假设评审编号为A、B、C，论文编号从0~9，可以任意得到一种分配方案为：A负责0~4，B负责5~9.（存在多种方案，这里只验证是否有分配方案的判断的正确性）。

输入相应的值，得到结果如下：



算法正确性得以验证。

1. 性能分析

固定论文数为500，评审数为100。

1. 固定a的值为50，增加b的值统计运行时间：



1. 固定b的值为30，增加a的值统计运行时间：



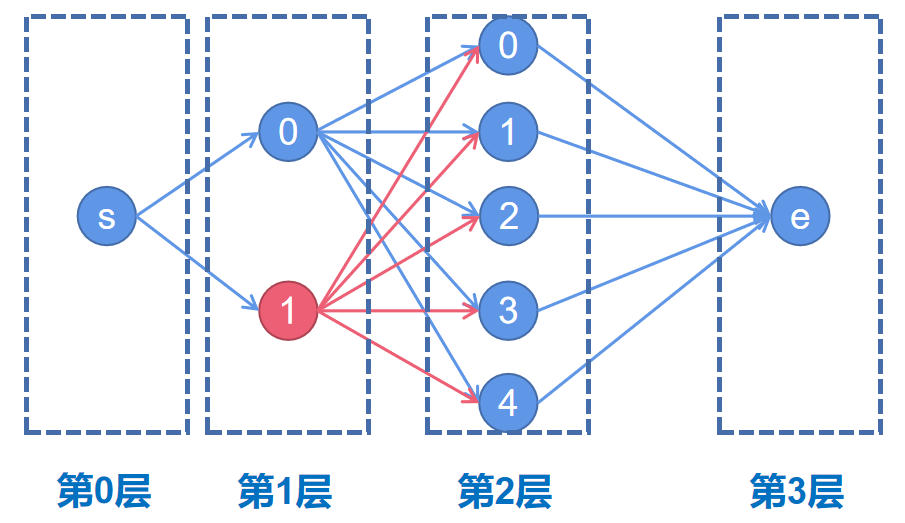
从上述运行时间统计结果可以知道，算法的实际运行时间与b的大小有关，运行时间会随b的增大而增大；当最大流的值保持不变时，运行时间又会随a的增大而减小。

**（四）Dinic算法**

1. 解题思路

该算法在上述算法的基础上，进行了分层和多路增广的优化，具体做法如下：

①使用BFS分层：采用BFS给每个节点标号，标号内容i为该点到源点经过的最短边数，根据标号内容i把原图构建为一个分层图，每条边均从第i层（所在层）连到第i+1层；（如下图所示）

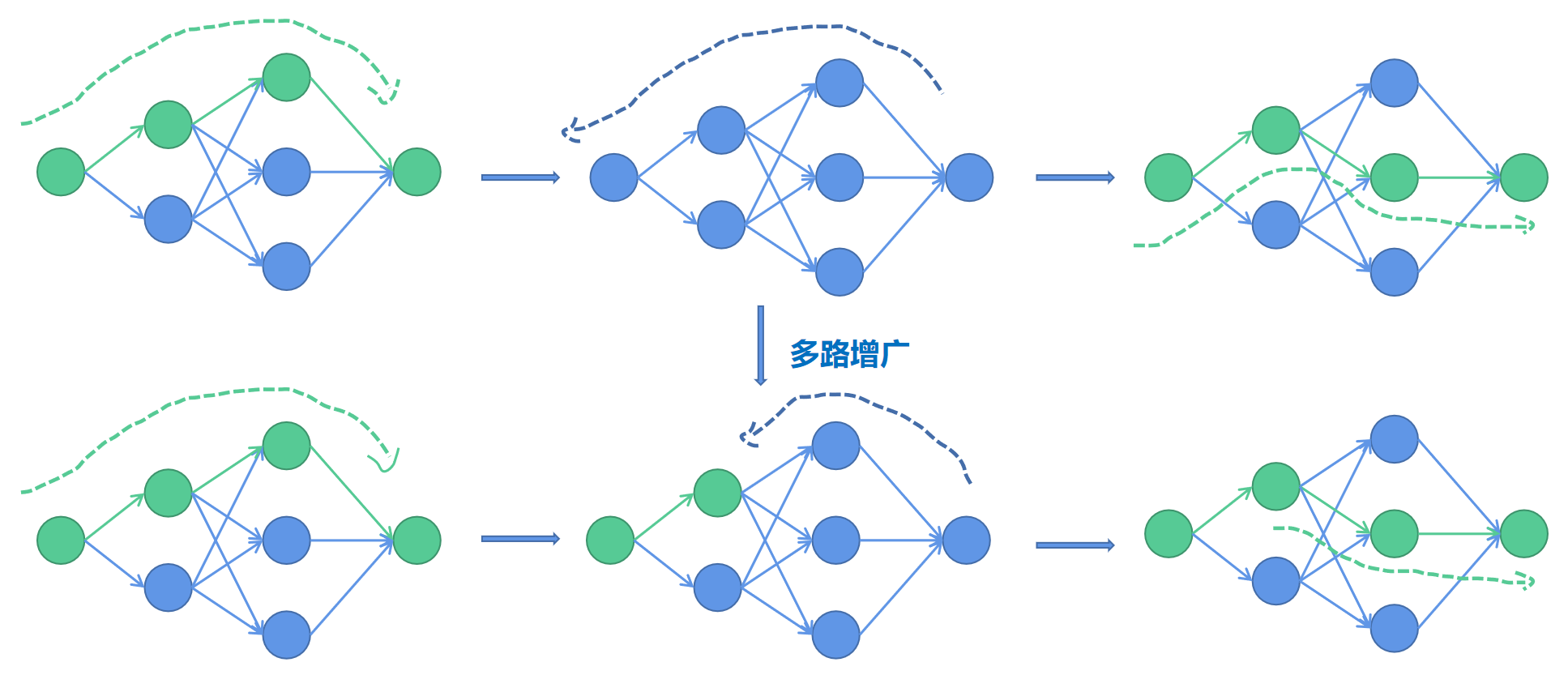


②DFS寻找增广路径：在分层图上使用DFS进行搜索，获得一条从s到e的路径，获得这条路径上的流量k（路径上残留最小的边的残量），将增广路径上正向边流量减去k，反向边流量加上k（引入多路增广思想，使一次DFS可以寻找多条增广路径）；

③重复上述过程，直到不再出现增广路径，获得最大流。

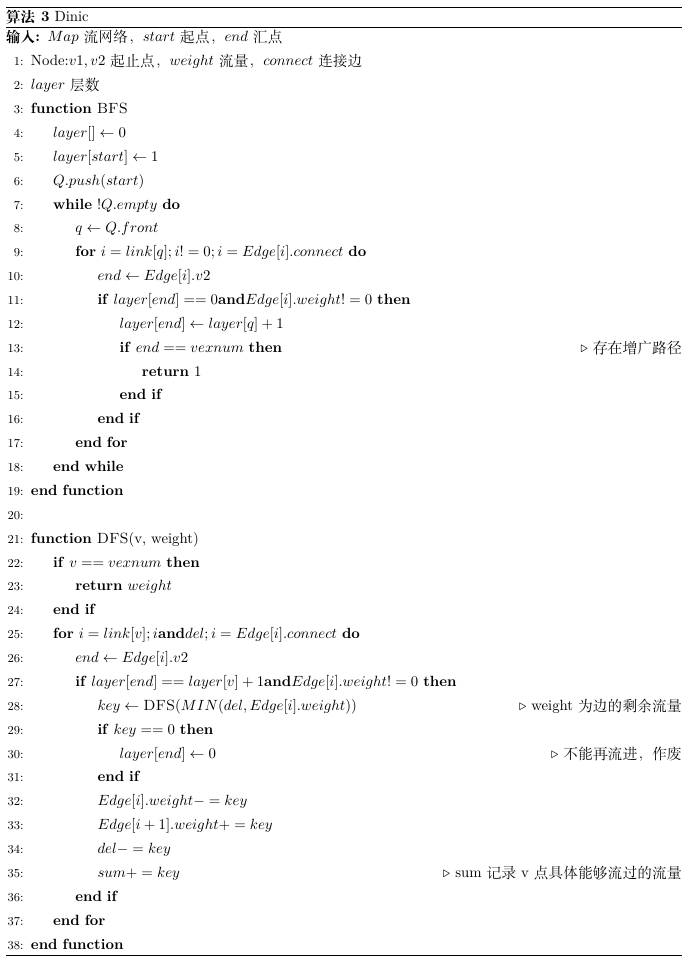
对**多路增广**的具体说明：

我们在使用DFS寻找增广路径时，找到一个增广路径以后就直接退回起点重新进行寻找。多路增广思想就是遍历一个增广路径后，只退回到上一个父亲节点，再往后寻找另外的增广路径，实现一次DFS寻找多条增广路径的效果。（如下图所示）



1. 伪代码

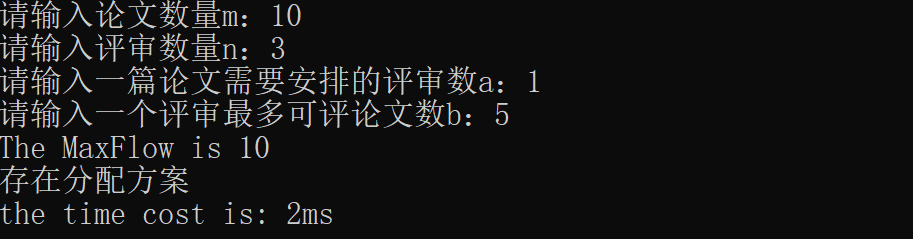
根据思路，编写每一个模块功能的伪代码如下：



1. 算法正确性验证

对于10篇论文、3个评委，将a赋值为1，b赋值为5。假设评审编号为A、B、C，论文编号从0~9，可以任意得到一种分配方案为：A负责0~4，B负责5~9.（存在多种方案，这里只验证是否有分配方案的判断的正确性）。

输入相应的值，得到结果如下：



算法正确性得以验证。

1. 性能分析

固定论文数为500，评审数为100。

1. 固定a的值为50，增加b的值统计运行时间：



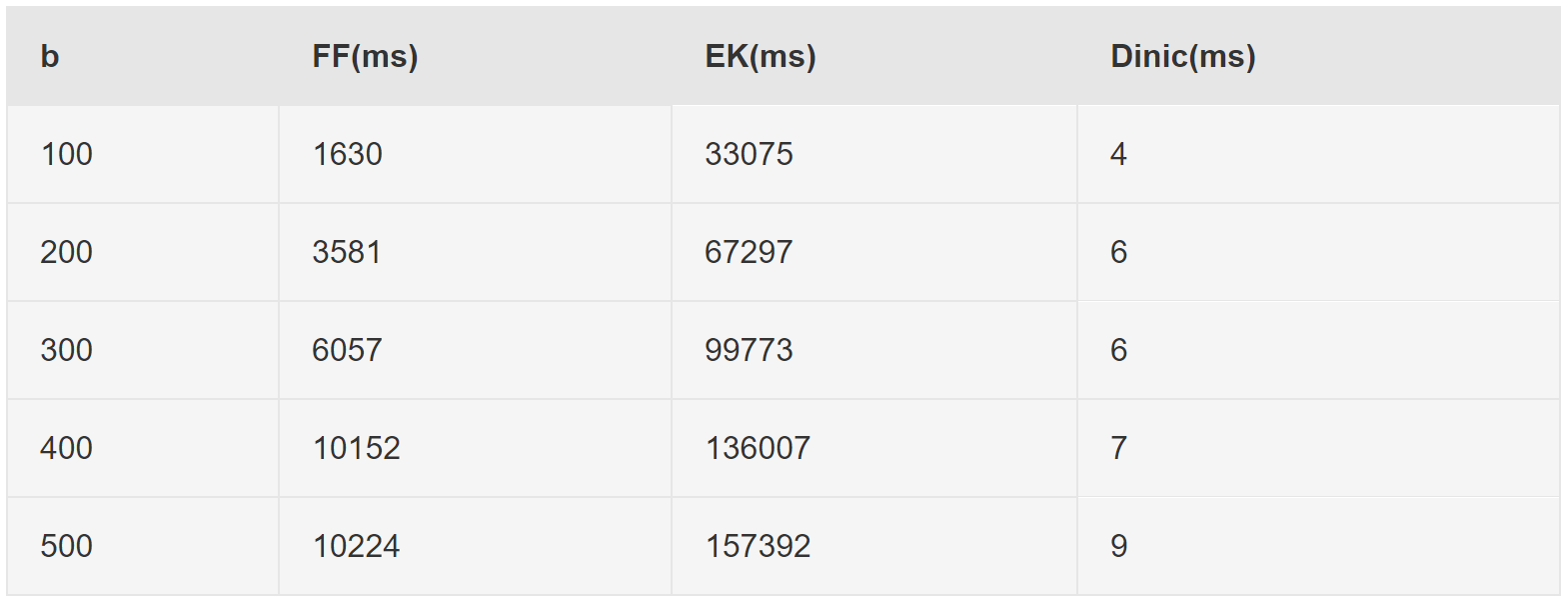
1. 固定b的值为30，增加a的值统计运行时间：



从上述运行时间统计结果可以知道，算法的实际运行时间与a、b的大小均有关，运行时间会随b的增大而增大，随a的增大而减小；当问题不存在分配方案时，运行时间变化趋于平缓。

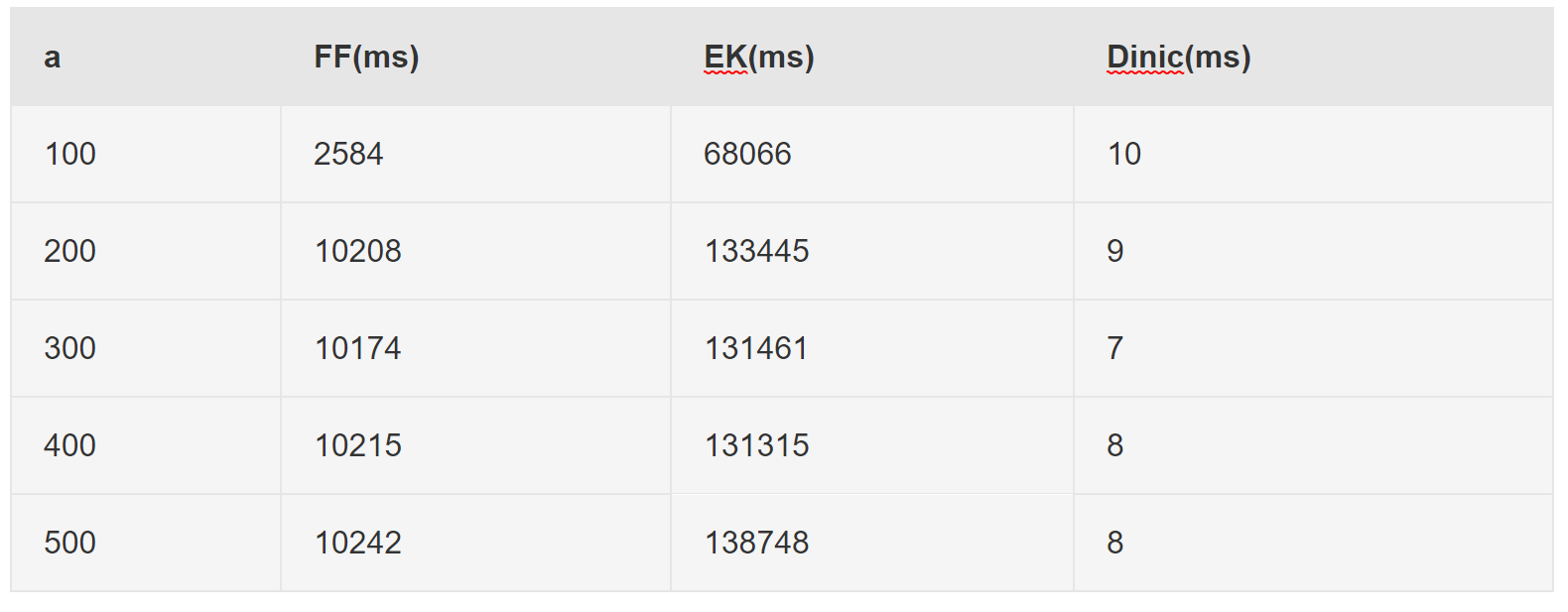
**（五）运行效率分析**

1. 固定m=400，n=200，a=200，增加b的值统计3种算法的运行时间：



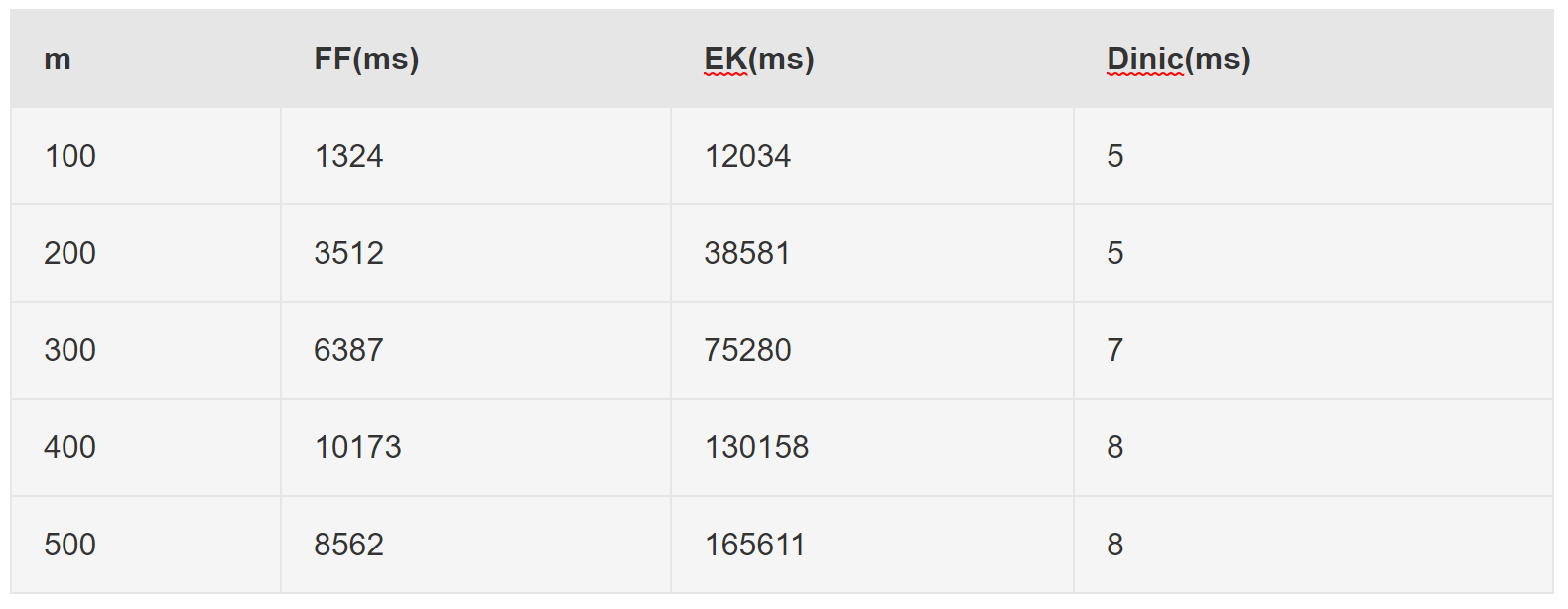
通过上述实验结果可以看到，3种算法的运行时间都随b的增大而增大，其中EK算法运行效率最低，运行时间明显高于另外两种算法。

1. 固定m=400，n=200，b=400，增加a的值统计3种算法的运行时间：



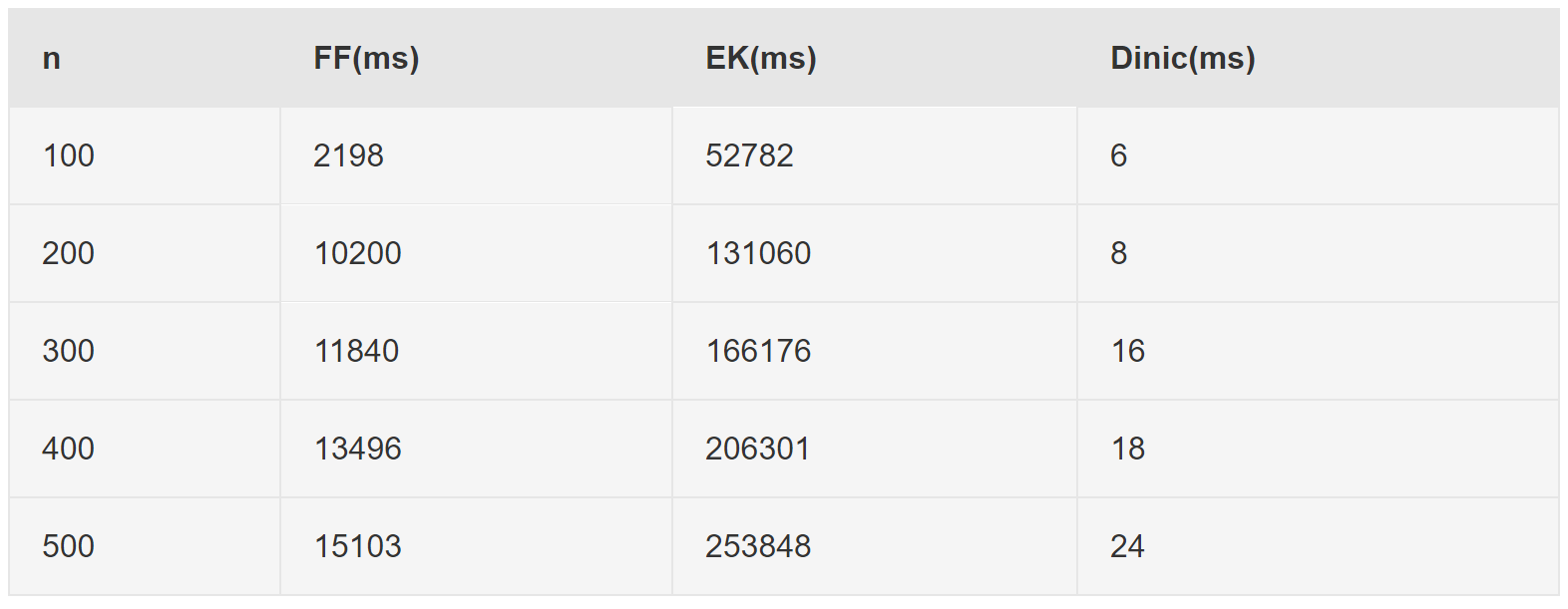
运行时间对a的增大会增大，但当分配方案不存在时，运行时间随a的变化幅度相比分配方案存在时的变化幅度小得多，基本保持不变。

1. 固定n=200，a=200，b=400，增加m的值统计3种算法的运行时间：



整体上，运行时间随m的增大而增大，EK算法的运行时间最长，且明显高于FF算法和Dinic算法，Dinic算法的运行效率最高。

1. 固定m=400，a=200，b=400，增加n的值统计3种算法的运行时间：



整体运行时间随n的增大而增大，算法的运行效率最高是Dinic，其次是FF，最低是EK。

1. 增加规模N，设定m=N，n=N/2，a=N/4，b=N，统计3种算法的运行时间：



从上述实验结果可以看到，EK算法的运行时间远远高于FF算法和Dinic算法，整体上算法的运行时间随规模的增大而增大。放大FF算法和Dinic算法的对比（如下图），可以看到在这两者中，Dinic的效率也显著高于FF算法。可以得出Dinic是一种可行的高效算法的结论。

# 四、实验结论或体会

该实验是利用最大流解决论文评审之间的分配问题。通过该实验学习了如何进行有效的问题转换，同时对FF方法下的EK算法以及Dinic算法进行了学习，通过实验对不同算法的运行效率进行分析，可以知道在实验采用的3种算法中，Dinic算法最高效。我们能也可以借助这一高效算法实现对其他可转换为最大流问题的实际问题进行解决。

|  |
| --- |
| 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。