**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： 图论——桥的查找**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 马里佳**

**报告人： 张欣杰 学号： 2020151091**

**实验时间： 2022/06/06**

**实验报告提交时间： 2022/06/26**

**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、实验原理：**   1. 一个医院有 名医生，现有 个公共假期需要安排医生值班。每一个公共假期由若干天（假日）组成，第 个假期包含的假日用 表示，那么需要排班的总假日集合为 。例如，“五一”假期由5月1日至5月7日一共7个假日组成。“元旦”假期由1月1日至1月3日一共3个假日组成。每名医生 可以值班的假日集合是 ，。例如，李医生可以值班的假日集合包括“五一”假期中的5月3日、5月4日和“元旦”假期中的1月2日。 2. 设计一个排班的方案使得每个假日都有一个医生值班并且满足下面两个条件：（1）每个医生最多只能值班个假日（2）每个医生在一个假期中只能值班1个假日。例如，安排李医生在“五一”假期中的5月4日值班 3. 根据上述场景完成下面任务：（1）实例化上述场景中的参数，生成数据；（2）设计一个多项式时间的算法求解上述问题，具体任务如下：基于生成的数据，设计一个流网络；解释说明该流网络中最大流与值班问题的解的关系；基于生成的数据，计算出排班的方案。 |
| **三、实验用品：**  操作系统：Windows 10  实验环境：Microsoft Visual Studio 2022 |
| **四、实验过程及内容：**   1. 问题分析   观察问题，通过问题分析可以将该值班问题化简为一张图，假设每一个医生为一个节点，每个节假日的每一天也为一个节点，若医生节点与节假日节点之间有通路的话，则可以表示该医生可以在该假日值班。将原图加上虚拟的源点和汇点，可以组成一张“流网络”用来表示值班的排班顺序。    由于条件的限制，因此我们需要对原图中加入限制，因此需要引入限流边和限流节点；限流节点限制的是每个假期值班的天数，在每个假期中限制流量为1；而限流边则是限制多人同时值班同一天的情况，将限流边和限流节点引入图中可以得到下图：    权值c为限制每个医生值班的天数，第二层为限制每个节假日只能值班一天，第三层和第四层限制的是同一天多人值班的情况；通过上图可以看到，当最大流大于等于总的假日天数的话，该图有合适的排班方案，否则没有合理的排班方案。   1. Ford-Fulkerson算法   Ford-Fulkerson算法是用来解决最大流算法的常用放大，FF算法通过不断寻找增广路径，然后让该路径中压入流并更新残留网络直到图中没有从源点到汇点的增广路径为止，即可得到图的最大流。  算法示例：    如图所示，该图的最大流为5  伪代码：    在本算法中，每一次运行dfs的时间为O(e)，最坏的情况为O(e\*f)。   1. Edmond-Karp算法   EK算法是基于FF算法的一个改进，该算法通过BFS寻找最短路径作为增广路径，然后以该路径做基准进行FF算法。算法实现思路：  (1)通过广度优先搜索（BFS）方法来寻找一条从源结点s到汇点t的增广路径。  (2)如果能找到，将该条增广路径上容量减去流的最小值作为图中当前流的一个增量。如果找不到，根据最大流最小切割定理可知，当前流就为最大流，此时Edmonds-Karp算法结束。  (3)该增广路径上的每一条边上的容量都减去这个增量，增广路径的反向边的容量都增加这个增量。  (4)重复上述步骤①②③，直到步骤①找不到增广路径，此时的流就为所求的最大流；伪代码：    算法复杂度，每一次寻找最短路径时间复杂度为O(e)，最多执行e\*v轮，因此最坏情况下为O(ve^2)   1. Dinic算法   首先对流网络使用bfs对节点进行分层标记，构建分层图，然后对分层图进行dfs生成增广路径，然后对该增广路径使用FF方法即可。  优化：(1)当前弧优化：在Dinic算法中，如果一条边已经增广满载了，那么改变即可以省略。但在DFS中每次找到一条增广路径后都是从头开始DFS，因此可以对已经满载的边进行标记，下一次DFS不需要遍历这条边。    (2)多路增广优化，在原算法中，每次找到一条增广路径都需要从源点开始重新dfs，但通过优化，可以将起始点回退到父节点进行搜索即可。  伪代码： |
| **五、实验现象及数据处理：**   1. 小规模数据测试   设置数据n=10，k=3，设置D和c的取值，记录在不同情况下的是否有解及运行时间的变化   |  |  |  |  |  | | --- | --- | --- | --- | --- | | D | c | RunTime/us | 最大流 | 是否有解 | | 2 | 10 | 29.4 | 6 | 有 | | 4 | 11 | 49.1 | 12 | 有 | | 6 | 12 | 117.8 | 18 | 有 | | 8 | 13 | 91.1 | 24 | 有 | | 10 | 14 | 107.9 | 30 | 有 | | 12 | 14 | 183.5 | 30 | 无解 |  1. 设置n=100，k=500，D=20，测试不用规模c下的运行时间/ms  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | c | 10 | 20 | 30 | 40 | 50 | | Ff | 11.407 | 24.280 | 63.613 | 115.538 | 198.404 | | Ek | 199.303 | 405.858 | 614.162 | 821.745 | 1043.79 | | Dinic | 0.411 | 0.472 | 0.483 | 0.506 | 0.564 |   上图为不通过规模的c数据下，Dinic算法是较快的，EK是较慢的，猜测是由于此时图比较稠密，而Ek算法O(ve^2)因此EK会较慢。   1. 设置参数为k=100，D=100，c=200，测试不同规模n的运行时间/ms   通过上图可以看到，在不同n的规模下，依旧是Dinic算法是最快的，FF次之，EK算法是最慢的，猜测是由于在EK进行广度优先搜索的时候，由于医生数量的增加，广搜过程会对假期进行下一次遍历的时候压入对列的时间增长从而导致总体时间增长。 |
| **六、实验结论：**  通过本次实验，我了解了解决最大流问题的三种算法，通过医生值班这一具体实例，加深了我对最大流的理解，同时也学会了如何使用具体实例搭建流网络，通过流网络来解决具体问题。在三种算法中，通过不同数据的测试，可以看到，在本问题中，Dicnic解决该问题的速度最快，其次是FF算法，最后是EK算法。从这几种增广路径中不难看出，图的遍历是解决图问题的最基本的方法，但是在不同情况下，也需要针对不同选择遍历图的方法。 |
| **思考题：** |
| **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**。