山东大学 软件 学院

众智科学与网络化产业 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202022300310 | 姓名： 张博宁 | | 班级： 软件20.7 |
| 实验题目：表决问题 | | | |
| 实验学时：3 | | 实验日期： 2022/6/10 | |
| 实验目的：  课程实验可以验证、巩固和补充课堂讲授的理论知识，是《众智网络和网络化产业》课程中一个重要的、不可或缺的实践环节。其目的是通过实验，使学生能够验证课程中所传授理论，并将理论联系实际解决问题。本课程本身就是一门交叉课程，课程实验可以训练学生利用本专业知识和所学习理论，综合运用经济学、社会学、计算与信息科学以及应用数学的有关概念与方法，讨论社会学和经济学中的计算思维，讨论各类互联网时代众智现象背后的内在机理或规律，培养学生用理工科的方法与工具来分析社会科学和经济学等问题，开阔学生视野，更好地面对未来的挑战。 | | | |
| 硬件环境：  处理器 AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz  机带 RAM 16.0 GB  系统类型 64 位操作系统, 基于 x64 的处理器 | | | |
| 软件环境：  Windows 11 家庭中文版  Embarcadero Dev-C++ Version 6.3  TDM-GCC 9.2.0 64-bit Release  Gephi 0.9 | | | |
| 实验步骤与内容：  **1、了解任务**  输入：给定m个人对n个项目按排序的投票。  输出：  1）确定其中是否隐含有孔多塞悖论（涉及到在有向图上尝试节点的拓扑排序）。  2）如果没有，就直接给出群体序，如果有，就按照一个特定的属性序，指出哪些投票是不满足单峰性质的，认为它们是“废票”，剔除后按照中位项定理给出群体排序。  **2、分析任务**  首先要保证m为奇数，此举保证了每一对项目均有一个少数服从多数的高下判断，继而可以建图进行拓扑排序。若拓扑排序成功则输出该唯一的拓扑序，算法结束。  然后根据中位项定理，先剔除部分数据，然后使用中位项定理的算法，即按照属性序（按照编号从小到大即可），然后选每个人最优的选项进行按属性序排列，每次从该排列中选取位置在中间的项目然后踢出即可。  **3、程序编写**  数据生成上，随机生成m个n的排列即可。具体操作时根据康托展开。对于每个表决人，生成一个1~n!的随机数s，代表这n个数的第s个排列，可使用c++ stl中的next\_permutation函数进行排列的生成：   1. int maxs=fact(N); 2. for(int i=0;i<M;i++){ 3. int s=rand()%maxs; 4. for(int j=0;j<N;j++)a[j]=j; 5. while(s--)next\_permutation(a,a+N); 6. for(int j=0;j<N;j++)printf("%d ",a[j]); 7. puts(""); 8. }   其中fact为阶乘函数，利用优化方法可以达到O((logn)^2)的复杂度，然而由于本项目数据规模不会太大，此处使用了O(n)的朴素算法。  群体表决的程序，先根据每个项目对，少数服从多数生成一个有向图：   1. for(int i=0;i<N-1;i++) 2. for(int j=i+1;j<N;j++){ 3. int big[2];big[0]=big[1]=0;*//big[0]:i is bigger* 4. for(int k=0;k<M;k++)big[rankof[k][i]>rankof[k][j]]++; 5. if(big[0]>big[1])addedge(i,j); 6. else addedge(j,i); 7. }   其中addedge(u,v)为在图中添加一条u->v的有向边：   1. inline void addedge(int u,int v){ 2. link[u].push\_back(v); 3. in[v]++; 4. }   此处link[]数组为邻接表。  然后进行拓扑排序并判断：   1. int sorted=0; 2. queue<int> res; 3. for(int i=0;i<N;i++) 4. if(!in[i])q.push(i); 5. while(!q.empty()){ 6. int o=q.front(); q.pop(); res.push(o); sorted++; 7. for(auto u:link[o]) 8. if(!--in[u])q.push(u); 9. } 10. if(sorted==N){ 11. printf("no Comdosette paradox!\norder:"); 12. while(!res.empty()){printf("%d ",res.front());res.pop();} 13. return 0; 14. }   若此处拓扑排序成功，输出排序时压入结果队列的拓扑序，程序结束。  排序不成功，进行程序的下一个流程。  中位项定理中，第一步需要筛选符合单峰性质的投票，以如下代码判断：   1. inline bool singlepeak(int x){ 2. int peak=vote[x][0]; 3. for(int i=peak+1;i<N;i++) 4. if(rankof[x][i]<rankof[x][i-1])return false; 5. for(int i=peak-1;i>=0;i--) 6. if(rankof[x][i]<rankof[x][i+1])return false; 7. return true; 8. }   如此枚举每一个投票人，对不符合条件的打上标记：   1. int cinvalid=0; 2. for(int i=0;i<M;i++) 3. if(!singlepeak(i)){invalid[i]++;printf("#%d ",i);cinvalid++;} 4. puts(""); 5. if(cinvalid==M){puts("no valid vote!");return 0;}   可见，若没有符合条件的投票，算法将会退出。  此时根据中位项定理的算法进行计算，直接模拟算法过程即可：   1. for(int ii=0;ii<N;ii++){ 2. int tong[N+5],srt[N+5],ite=0; 3. for(int i=0;i<N;i++)tong[i]=0; 4. for(int j=0;j<M;j++){ 5. if(invalid[j])continue; 6. for(int k=0;k<N;k++)if(!picked[vote[j][k]]) 7. {tong[vote[j][k]]++;break;} 8. } 9. for(int i=0;i<N;i++)if(tong[i])srt[++ite]=i; 10. int mid=(ite>>1)+1; 11. printf("%d ",srt[mid]); 12. picked[srt[mid]]++; 13. }   从最大嵌套数的循环来看，时间复杂度O(MN^2)，即可算出根据中位项定理得出的群体排序。  **4、调试与结果分析**  为了避免过多矛盾，随机生成的数据首先使用M=N=3计算，结果如下：  ，基于此的计算结果：    可见此次没有出现孔多塞悖论，直接通过拓扑排序完成了群体排序。  增大数据范围，调整为M=7,N=5：  计算结果：    可见仍然没有出现孔多塞悖论，基于此数据范围继续生成：  计算结果：    出现了孔多塞悖论，经过筛选后仅一组数据满足单峰性质，经中位数算法后结果与该人的投票结果一致。  继续增大数据范围到M=21，N=7：  计算结果：    可见如此大的数据范围下，仍然有可能不出现孔多塞悖论。  继续生成数据：  结果：    出现了孔多塞悖论，但符合没有人的投票符合单峰性质，此种结果出现概率极大，说明随机的条件下很难生成符合单峰性质的数据。  将范围改为M=21，N=5：  结果：    可见程序根据留下的几个单峰数据进行了群体排序。 | | | |
| 结论分析与体会：  经过本次实验，我熟悉了利用代码研究社会问题的方式与模拟算法实现的书写。熟悉了孔多塞悖论、中位项定理的内容与判断方法。 | | | |

实验目录下仍然有生成、运算的代码，编译后运行即可。