

**实 验 报 告**

**（2019 / 2020 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 离散数学 | | | | | |
| 实验名称 | 图的随机生成及欧拉(回)路的确定 | | | | | |
| 实验时间 | 2019 | 年 | 11 | 月 | 25 | 日 |
| 指导单位 | 计算机学院计算机科学与技术系 | | | | | |
| 指导教师 | 陈兴国 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 吴雯 | 班级学号 | B18030322 |
| 学院(系) | 计算机学院 | 专 业 | 计算机科学与技术系 |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | 图的随机生成及欧拉(回)路的确定 | | | **指导教师** | 陈兴国 |
| **实验类型** | **综合** | **实验学时** | **4** | **实验时间** | 2019.11.25 |
| 1. **实验目的和要求**   **实验目的：**  **编程随机生成n个结点的无向图并能进行（半）欧拉图的判定，若是则给出欧拉（回）路。**  **实验要求：**  对给定n个结点，随机生成邻接矩阵以确定某无向简单图并进行欧拉图和半欧拉图的判定，若符合则给出至少一条欧拉回路或欧拉路。  程序的能够根据输入的n和m，随机生成具有n个结点m个边的简单无向图（能够判断n和m的合理性），然后判断图的连通性，如果这个图是个连通图，再计算图中度数是奇数的结点个数，判断是欧拉图还是半欧拉图，如果是欧拉图或者半欧拉图，然后根据输入打印一个欧拉(回)路,或者所有的欧拉(回)图。 | | | | | |
| 二、**实验环境(实验设备)**  硬件：微型计算机  软件：Windows 操作系统、Microsoft Visual C++6.0、Java等可视化编程语言 | | | | | |
| **三、实验原理及内容**  说明：这部分内容主要包括：  1、形式化描述实验中所使用的数据结构和存储结构，给出函数之间的调用关系和数据传递方式；  1.1随机生成图 void Generate() //随机生成图  1.2判断(半)欧拉图 int Judge() //判断(半)欧拉图  1.3判断连通性 int JudgeLianTong() //判断连通性  1.4寻路 void FindLu(int cur)  1.5主函数 main()  2、给出核心算法的C++或Java等语言的源代码，并加上详细注释，分析算法的时间复杂度；  2.1  void Generate() //随机生成图  {  printf("正在生成%d个结点%d条边的简单无向图......\n", n, m);  int cnt = 0;  srand(time(NULL));  while (cnt < m)  {  int x = rand() % n;  int y = rand() % n;  if (x != y && G[x][y] == 0)  {  G[x][y] = 1;  G[y][x] = 1;  cnt++;  }  }  printf("生成完成\n\n");  if (n <= 10)  {  printf("图的邻接矩阵为:\n");  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  printf("%d ", G[i][j]);  }  printf("\n");  }  printf("\n");  }  }  2.2判断(半)欧拉图 int Judge() //判断(半)欧拉图  int Judge() //判断(半)欧拉图  {  int flag = 0;  for (int i = 0; i < n; i++)  {  int cnt = 0;  for (int j = 0; j < n; j++)  {  if (G[i][j] == 1)  {  cnt++;  }  }  if (cnt % 2 == 1)  {  flag++;  }  }  if (flag == 0)  {  return 0; //欧拉回路  }  else if (flag == 2)  {  return 1; //欧拉路  }  else  {  return -1; //不是欧拉路也不是欧拉回路  }  }  2.3判断连通性 int JudgeLianTong() //判断连通性  int JudgeLianTong() //判断连通性  {  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  P[i][j] = G[i][j];  T[i][j] = G[i][j];  }  }  for (int k = 2; k <= n; k++) //n的4次方复杂度,计算可达性矩阵  {  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  int t = 0;  for (int a = 0; a < n; a++)  {  t += T[i][a] \* G[a][j];  }  if (t == 0)  {  TT[i][j] = 0;  }  else  {  TT[i][j] = 1;  }  }  }  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  T[i][j] = TT[i][j];  }  }  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  if (T[i][j] > 0 || P[i][j] > 0)  {  P[i][j] = 1;  }  }  }  }  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  if (i != j && P[i][j] == 0)  {  return 0;  }  }  }  return 1;  }  2.4寻路 void FindLu(int cur)  void FindLu(int cur)  {  if (choice == 1 && has == 1) return;  if (cnt == m + 1)  {  for (int i = 0; i < cnt; i++)  {  if (i == 0) printf("%d", record[i]);  else printf("->%d", record[i]);  }  printf("\n");  has = 1;  }  else  {  for (int i = 0; i < n; i++)  {  if (G[cur][i] == 1 && vis[cur][i] == 0)  {  vis[i][cur] = vis[cur][i] = 1;  record[cnt++] = i;  FindLu(i);  cnt--;  vis[i][cur] = vis[cur][i] = 0;  }  }  }  }  2.5主函数  int main()  {  do  {  printf("请输入无向图的结点个数:");  scanf\_s("%d", &n);  printf("请输入边的个数:");  scanf\_s("%d", &m);  if (m > n\*(n - 1) / 2)  {  printf("%d个结点的无向图最多有%d条边\n", n, n\*(n - 1) / 2);  }  } while (m > n\*(n - 1) / 2); //判断n和m的合理性  Generate(); //随机生成图  if (JudgeLianTong() == 0) //判断连通性  {  printf("这个图不是一个连通图，所以也不是欧拉图和半欧拉图\n");  }  else  {  printf("\n这个图是连通图\n");  int tmp = Judge(); //判断(半)欧拉图  if (tmp == 0)  {  printf("这个图是一个 欧拉图\n");  printf("-----------------1.打印一个欧拉回路------------------\n");  printf("-----------------2.打印所有欧拉回路------------------\n");  printf(" 输入你的选择:");  scanf\_s("%d", &choice);  if (choice == 1)  {  printf("其中一条欧拉回路为:\n");  record[cnt++] = 0;  FindLu(0); //找出回路  cnt--;  }  else if (choice == 2)  {  printf("所有的欧拉回路为:\n");  for (int i = 0; i < n; i++)  {  record[cnt++] = i;  FindLu(i);  cnt--;  }  }  else  {  printf\_s("-----------------输入有误-----------------\n");  system("pause");  }  }  else if (tmp == 1)  {  printf("这个图是一个 半欧拉图\n");  printf("-----------------1.打印一个欧拉路------------------\n");  printf("-----------------2.打印所有欧拉路------------------\n");  printf(" 输入你的选择:");  scanf\_s("%d", &choice);  if (choice == 1)  {  for (int i = 0; i < n; i++)  {  int t = 0;  for (int j = 0; j < n; j++)  {  if (G[i][j] == 1)  {  t++;  }  }  if (t % 2 == 1)  {  record[cnt++] = i;  FindLu(i);  cnt--;  break;  }  }  }  else if (choice == 2)  {  printf("所有的欧拉路为:\n");  for (int i = 0; i < n; i++)  {  int t = 0;  for (int j = 0; j < n; j++)  {  if (G[i][j] == 1)  {  t++;  }  }  if (t % 2 == 1)  {  record[cnt++] = i;  FindLu(i);  cnt--;  }  }  }  else  {  printf("-----------------输入有误-----------------\n");  }  }  else  {  printf("这个图既不是欧拉图，也不是半欧拉图\n");  }  }  system("pause");  return 0;  }  3、给出测试数据及运行结果、实验相关结论等。  3.1    3.2 | | | | | |

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **四、实验小结**（包括问题和解决方法、心得体会、意见与建议等）  中文五号宋体，英文五号Times new roman字体，1.25倍行距  **说明：这部分内容主要包括：在编程、调试或测试过程中遇到的问题及解决方法、本次实验的心得体会、进一步改进的设想等。**  (一)实验中遇到的主要问题及解决方法  如：在程序运行时，总出现…问题，后来用…方法查找，发现是…所导致的错误，进行…的修改后错误消除，从而说明……  在判断是欧拉图还是半欧拉图时，一开始没有判断是不是连通图，也不知道判断是否只存在零个或者两个奇数度结点，后来通过查阅资料，复习基础知识才知道要先判断是不是连通图，要判断有是否只存在零个或者两个奇数度结点，两个则是半欧拉图，零个则是欧拉图，从而说明基础知识也要牢牢掌握；  在输出欧拉路的时候，总是输出的问题，后来通过利用递归深度搜索逆序输出结点，确保找到一条完整的路径，避免存在回路没有被遍历到，解决辽问题，说明递归也是一个很重要的思想  （二）实验心得  这次离散实验让我对C++的编程语言有了更强的运用能力，尤其是继承与派生的知识有了更深刻的理解，让我明白了在平时的学习中要勤查资料，多学习一些课外知识。离散数学和编程知识、数据结构是息息相关、密不可分的:以后我将更加认真学习离散数学，并且更多地将编程的知识运用起来，提升自己的实际运用能力。  （三）意见与建议（没有可省略） | | | | | |
| 1. **支撑毕业要求指标点**   支撑毕业要求的指标点为：  √   * 1-4掌握计算机科学与技术领域的专业知识，能将专业知识用于分析和解决计算机领域复杂工程问题。 * 2-1能够应用数学、自然科学和工程科学的基本知识，识别和分析计算机领域复杂工程问题的特征。 | | | | | |
| **六、指导教师评语 (含学生能力达成度的评价)** | | | | | |
| **成 绩** |  | **批阅人** |  | **日 期** |  |

如果不太想写太多字，“指导教师评语”也可以设计为如下的各选择项用打勾形式（仅仅作为一个简单示例，请各课程负责人根据课程和实验情况以及支撑的指标点来自行设定选择项，同一门课程的不同实验评分细则项允许存在不同）：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **评 分 细 则** | **评分项** | **优秀** | **良好** | **中等** | **合格** | **不合格** |
| **遵守实验室规章制度** |  |  |  |  |  |
| **学习态度** |  |  |  |  |  |
| **算法思想准备情况** |  |  |  |  |  |
| **程序设计能力** |  |  |  |  |  |
| **解决问题能力** |  |  |  |  |  |
| **课题功能实现情况** |  |  |  |  |  |
| **算法设计合理性** |  |  |  |  |  |
| **算法效能评价** |  |  |  |  |  |
| **回答问题准确度** |  |  |  |  |  |
| **报告书写认真程度** |  |  |  |  |  |
| **内容详实程度** |  |  |  |  |  |
| **文字表达熟练程度** |  |  |  |  |  |
| **其它评价意见** |  | | | | |
| **本次实验能力达成评价（总成绩）** |  |  |  |  |  |