

**《数据结构与算法实践》实践报告**

|  |
| --- |
| 学 院： 虚拟现实学院 。  年级 专业 班级：18级 软件工程 APP开发技术1班  课程（环节）名称： 《数据结构与算法实践》 。 |
| 学 生 学 号： 11827041 。  学 生 姓 名： 叶欣宇 。  指 导 教 师： 张亮 。 |

2019年6月24日

|  |  |
| --- | --- |
| **实践题目** | 弗洛依德应用实践 |
| **实践时间** | 2019.6.24-2019.6.28 |
| **一、实践目的**  1.熟练应用弗洛依德最短路径方法。  2.熟练掌握弗洛依德最短路径编程思想。  3.能够基于弗洛依德最短路径编程思想形式下，设计合理算法解决最短路径应用问题。 | |
| **二、实践内容**  基于弗洛依德最短路径编程思想形式下，设计合理算法，根据图中的有向边<vi，vj>表示第i个村庄和第j个村庄之间有道路，边上的权表示这条道路的长度，利用弗洛依德最短路径原理，求出任意两个村庄间的最短路径。  。 | |
| **三、实践步骤**  （一）算法的伪代码  1.建立有向图G；  1.1确定图的顶点个数vertexNum和边的个数edgeNum；  1.2输入顶点信息存储在一维数组edge[MaxSize][MaxSize]中；  1.3初始化邻接矩阵，用INFINITY存储一个正无穷值，设此正无穷值为6666，如果i=j则路径为0，如果i！=j，路径为INFINTITY；   1. 输入村庄之间的路径以及距离，存入邻接矩阵； 2. 用弗洛伊德算法求最短路径；   3.1考虑路径vi,v0,vj是否存在，如果存在，则比较vi,vj和vi,v0,vj的路径长度，取较短者为从vi到vj的中间顶点的序号不大于0的最短路径。  3.2在路径上再增加一个顶点v1，依此类推，在经过n次比较后，最后求得的必是从顶点vi到顶点vj的最短路径；  4.输出最后的邻接矩阵即任意两个村庄间的最短路径。  （二）算法的C语言描述  void Floyd(MGraph \*G,int n,int m){  for(int k=1;k<=G->vertexNum;k++)  for(int i=1;i<=G->vertexNum;i++)  for(int j=1;j<=G->vertexNum;j++)  if(G->edge[i][j]>G->edge[i][k]+G->edge[k][j] )  G->edge[i][j]=G->edge[i][k]+G->edge[k][j];  }  （三）时间复杂度分析  问题规模为G->vertexNum，即村庄的个数n，  算法核心语句为：  for(int k=1;k<=G->vertexNum;k++)  for(int i=1;i<=G->vertexNum;i++)  for(int j=1;j<=G->vertexNum;j++)  if(G->edge[i][j]>G->edge[i][k]+G->edge[k][j] )  G->edge[i][j]=G->edge[i][k]+G->edge[k][j];  故时间复杂度为O(n3)  （四）编译  编译通过。  （五）测试 | |
| **四、实践项目成果**  //  // main.c  // Floyd  //  // Created by 叶欣宇 on 2019/6/27.  // 版权所有，盗版必究！  //  #include <stdio.h>  #define MaxSize 10  #define INFINITY 6666  int t1,t2,t3;  typedef struct{  int edge[MaxSize][MaxSize];  int vertexNum,edgeNum;  }MGraph;  void CreatGraph(MGraph \*G,int n,int m){  G->vertexNum = n;  G->edgeNum = m;  for(int i=1;i<=G->vertexNum;i++)  for(int j=1;j<=G->vertexNum;j++)  if(i==j) G->edge[i][j]=0;  else G->edge[i][j]=INFINITY;  }  void Floyd(MGraph \*G,int n,int m){  for(int k=1;k<=G->vertexNum;k++)  for(int i=1;i<=G->vertexNum;i++)  for(int j=1;j<=G->vertexNum;j++)  if(G->edge[i][j]>G->edge[i][k]+G->edge[k][j] )  G->edge[i][j]=G->edge[i][k]+G->edge[k][j];  }  void min(MGraph \*G,int n,int m){  int min = 6666;  int ii = 0,jj = 0;  for(int i=1;i<=G->vertexNum;i++){  for(int j=1;j<=G->vertexNum;j++){  if(min>G->edge[i][j]&&G->edge[i][j]!=0){  ii=i;  jj=j;  min=G->edge[i][j];  }  }  }  printf("%d村庄到%d村庄距离最短，为：%d\n",ii,jj,min);  }  int main(){  MGraph G;  int n,m;  printf("请输入村庄的个数，以及道路条数：");  scanf("%d %d",&n,&m);  CreatGraph(&G,n,m);    printf("输入两个村庄之间道路的距离:\n");  for(int i=1;i<=G.edgeNum;i++){  scanf("%d %d %d",&t1,&t2,&t3);  G.edge[t1][t2]=t3;  }  printf("村庄间道路的距离:\n");  for(int i=1;i<=G.vertexNum;i++){  for(int j=1;j<=G.vertexNum;j++){  printf("%8d",G.edge[i][j]);  }  printf("\n");  }    Floyd(&G,n,m);  printf("村庄间道路最短的距离:\n");  for(int i=1;i<=G.vertexNum;i++){  for(int j=1;j<=G.vertexNum;j++){  printf("%5d",G.edge[i][j]);  }  printf("\n");  }  min(&G, n, m);  return 0;  } | |
| **五、教师评语**  **报告成绩： 教师签字：** | |