**理工学院计算机类课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 数据结构 | | 实验名称 | 实验6 Project 图 |
| 小组名称及成员 | 吴警 | | | |
| 专 业 | 软件工程 | | 年 级 | 2019 |
| 班 级 |  | | 实验时间 | 第11周 |
| 教师评语（或成绩） | | 教师签字：  年 月 日 | | |
| **一、实验目的**  （1）了解C项目编写方式  （2）熟悉图的基本运算以及算法与程序之间的关系 | | | | |
| **二、实验设备（工具、材料、硬软件）及要求**  1、设备：计算机一台、Dev C++  2、具体要求： 利用图结构，模拟安卓手机锁屏图案密码设计（即在9个点中一笔画4条边）。  各小组可根据实际情况，使用邻接矩阵或邻接表创建图，然后实现相应的DFS或BFS算法。 | | | | |
| **三、实验过程**  要求：（1）列出实现的源代码、运行效果截图；（2）为源码添加必要的注释语句。另外，对实验进一步分析存在的问题提出解决办法，或总结实验所取得的经验均可写入报告中。**能力强者也可使用java语言实现代码。**  **源代码：**  **main.c：**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include"graph.h"**  **int main(int argc, char \*argv[]) {**  **int x;**  **printf("请选择以下操作:1 \*DFS\* or 2 \*DFSTraversal\*\n");**  **scanf("%d",&x);**  **MGraph G;**  **G=createMGraph(G);**  **if(x==1)**  **DFS(G,0);**  **else if(x==2)**  **DFSTraversal(G);**    **return 0;**  **}**  **图结构体及函数：**  **#include<stdio.h>**  **#define MAXVEX 100**  **typedef int VertexType;**  **typedef int EdgeType;**  **struct MGraph{**  **VertexType vertex[MAXVEX]; //顶点**  **EdgeType edge[MAXVEX][MAXVEX]; //边**  **int vexnum,edgenum; //v，顶点数量：：e，边数量**  **};**  **typedef struct MGraph MGraph;**  **//使用邻阶矩阵 创建图**  **MGraph createMGraph(MGraph G){**  **//输入顶点个数，边个数**  **printf("vexnum = ");**  **scanf("%d",&G.vexnum);**  **printf("edgenum = ");**  **scanf("%d",&G.edgenum);**  **//输入顶点信息**  **int i,j,k;**  **printf("vertex:\n");**  **for(i=0;i<G.vexnum;i++)**  **{**  **scanf("%d",&G.vertex[i]);**  **}**  **for(i=0;i<G.vexnum;i++)**  **for(j=0;j<G.vexnum;j++)**  **G.edge[i][j]=0; //初始化 边**  **printf("edge:\n");**  **for(k=0;k<G.edgenum;k++){//逐一输入边的信息**  **scanf("%d%d",&i,&j); //输入有边的顶点i和j**  **G.edge[i][j]=1;**  **//G.edge[j][i]=G.edge[j][i]; //只用于无向图**  **}**  **return G;**  **}**  **int visited[MAXVEX]={0};**  **//使用邻接矩阵存储的DFS算法**  **void DFS(MGraph G,VertexType v){//从顶点v开始，访问图G**    **//visited数组是顶点是否被访问过 ，初始值0代表未被访问**    **visited[v]=1;//标记顶点v被访问，被访问值改为1**  **printf("%5d",G.vertex[v]);//输出当前 被访问的顶点信息**  **int i;**  **for(i=0;i<G.vexnum;i++){**  **//顶点i和顶点v之间有边，且 顶点i没有被访问过**  **if((G.edge[v][i]==1)&&(visited[i]==0)){**  **DFS(G,i);//递归调用，从下标为i的顶点开始继续DFS**  **}**  **}**  **}**  **void DFSTraversal(MGraph G){**  **int i;**  **for(i=0;i<G.vexnum;i++){**  **if(visited[i]==0)**  **DFS(G,i);**  **}**  **}**  **运行结果：**  **capture_20200518220223558**  **实验过程中遇到的问题：**  **·代码理解不透彻，需要参考老师的代码才能完成实验。**  **·不能够灵活运用图的算法。**  **改进措施：**  **·通过CSDN等软件学习图的相关知识。**  **·加强对图的算法相关代码的理解。**  **收获：**  **·学会了用邻阶矩阵矩阵表示图。**  **·了解了一些图的算法**  **BFS 常用于找单一的最短路线，它的特点是 “搜到就是最优解”；**  **DFS 常用于找所有解的问题，找到的不一定是最优解。。**  **·对图的搜索有了进一步的了解** | | | | |