**Project 8 邻接矩阵和邻接表**

学号 2014211116 姓名 骆金参 上交时间 2015-11-29

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题号 | 程序逻辑(40) | 算法新颖性(20) | 代码规范 (20) | 实验报告(20) | 总分 |
| 得分 |  |  |  |  |  |

1. **问题描述**

邻接矩阵和邻接表的创建和输出

**2. 算法描述**

**/\* 键盘输入顶点和边的信息，建立图的邻接矩阵表示 \*/**

**// Exercise 1: Create an adjacency matrix of a directed graph**

**void CreateMGraph(MGraph \*G)**

**{**

**int i,j,k,w;**

**printf("输入顶点数和边数:\n");**

**scanf("%d %d",&G->numNodes,&G->numEdges); /\* 输入顶点数和边数 \*/**

**for(i = 0;i <G->numNodes;i++) /\* 读入顶点信息,建立顶点表 \*/**

**scanf(&G->vexs[i]);**

**for(i = 0;i <G->numNodes;i++)**

**for(j = 0;j <G->numNodes;j++)**

**G->arc[i][j]=0; /\* 邻接矩阵初始化 \*/**

**printf("输入边(vi,vj)上的下标i，下标j和权w:\n");**

**for(k = 0;k <G->numEdges;k++) /\* 读入numEdges条边，建立邻接矩阵 \*/**

**{**

**scanf("%d %d %d",&i,&j,&w); /\* 输入边(vi,vj)上的权w \*/**

**G->arc[i][j]=w;**

**}**

**}**

**/\* 用图的邻接矩阵输出边的信息，如边的头和尾及权值 \*/**

**// Exercise 2: Output the edges' tail and head and weight**

**void OutPutMGraph(MGraph \*G)**

**{**

**int i,j;**

**for(i=0;i<G->numNodes;i++)**

**{**

**for(j=0;j<G->numEdges;j++)**

**printf("%6d",G->arc[i][j]);**

**printf("\n");**

**}**

**}**

**/\* 键盘输入顶点和边的信息，建立图的邻接表结构 \*/**

**// Exercise 1: Create an adjacency list of a directed graph**

**void CreateALGraph(GraphAdjList \*G)**

**{**

**int i,j,k;**

**EdgeNode \*e;**

**printf("输入顶点数和边数:\n");**

**scanf("%d %d",&G->numNodes,&G->numEdges); /\* 输入顶点数和边数 \*/**

**for(i = 0;i < G->numNodes;i++) /\* 读入顶点信息,建立顶点表 \*/**

**{**

**scanf(&G->adjList[i].data); /\* 输入顶点信息 \*/**

**G->adjList[i].firstedge=NULL; /\* 将边表置为空表 \*/**

**}**

**for(k = 0;k < G->numEdges;k++)/\* 建立边表 \*/**

**{**

**printf("输入边(vi,vj)上的顶点序号:\n");**

**scanf("%d %d",&i,&j); /\* 输入边(vi,vj)上的顶点序号 \*/**

**e=(EdgeNode \*)malloc(sizeof(EdgeNode)); /\* 向内存申请空间,生成边表结点 \*/**

**e->adjvex=j; /\* 邻接序号为j \*/**

**e->next=G->adjList[i].firstedge; /\* 将e的指针指向当前顶点上指向的结点 \*/**

**G->adjList[i].firstedge=e; /\* 将当前顶点的指针指向e \*/**

**e=(EdgeNode \*)malloc(sizeof(EdgeNode)); /\* 向内存申请空间,生成边表结点 \*/**

**e->adjvex=i; /\* 邻接序号为i \*/**

**e->next=G->adjList[j].firstedge; /\* 将e的指针指向当前顶点上指向的结点 \*/**

**G->adjList[j].firstedge=e; /\* 将当前顶点的指针指向e \*/**

**}**

**}**

**/\* 用图的邻接表结构输出边的信息，如边的头和尾及权值 \*/**

**// Exercise 2: Output the edges' tail and head and weight**

**void OutPutALGraph(GraphAdjList \*G)**

**{**

**int i = 0;**

**while(G->adjList[i].firstedge != NULL && i < MAXVEX)**

**{**

**printf("顶点%d:%c ", i,G->adjList[i].data);**

**EdgeNode \*e = NULL;**

**e = G->adjList[i].firstedge;**

**while(e != NULL)**

**{**

**printf("%d->", e->adjvex);**

**e = e->next;**

**}**

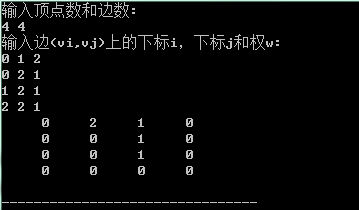
**i++;**

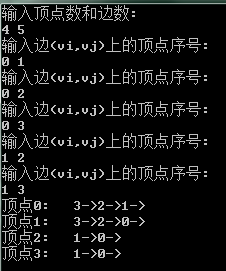
**printf("\n");**

**}**

**}**

**3.测试结果**





**4. 分析与评论**

没有提供测试数据，感觉很混乱、、、