实验三 串和数组

学院 专业 班

学号 姓名

一．实验目的

1．了解串的定义及特点；

2．掌握串的模式匹配算法；

3．掌握稀疏矩阵的三元组存储表示；

4. 掌握三元组的快速转置算法；

二．实验内容

1. 实现字符串的模式匹配算法（下标从0开始）

2. 实现三元组的快速转置算法。

三．实验步骤

1. （1） 程序代码

（2）运行程序

（3）运行结果（截图）

2. （1） 程序代码

（2）运行程序

（3）运行结果（截图）

四．实验小结

三元组的快速转置算法 （参考程序 ，程序填空）

#include<stdio.h>

#include<malloc.h>

#include "stdlib.h"

#define MAXSIZE 1000

typedef int ElemType;

typedef struct {

int i, j; //该非零元的行下标和列下标

ElemType v; // 该非零元的值

} Triple; // 三元组类型

typedef struct {

Triple data[MAXSIZE + 1];

int mu, nu, tu;

} TSMatrix;

void FastTransposeSMatrix(TSMatrix &M, TSMatrix &T) {

// 采用三元组顺序表存储表示，求稀疏矩阵M的转置矩阵T

int col, t, p, q,k;

int num[20], cpot[20];

for(k=1;k<=M.tu;k++)

{

printf("%3d%3d%3d\n",M.data[k].i,M.data[k].j,M.data[k].v);

}

T.mu= M.nu; T.nu= M.mu; T.tu=M.tu;

if (T.tu) {

for (col=1; col<=M.nu; ++col) num[col] = 0;

for (t=1; t<=M.tu; ++t) // 求 M 中每一列所含非零元的个数

++num[M.data[t].j];

cpot[1] = 1;

// 求 M 中每一列的第一个非零元在 T.data 中的序号

for (col=2; col<=M.nu; ++col)

cpot[col] = [1]

printf("\n转置前cpot:\n");

for (col=1; col<=M.nu; ++col)

printf("%3d", cpot[col]) ;

for (p=1; p<=M.tu; ++p) {

col = M.data[p].j;

q = cpot[col];

自己填 [2] [3][4][5]

} // for

printf("\n转置后cpot:\n");

for (col=1; col<=M.nu; ++col)

printf("%3d", cpot[col]) ;

} // if

} // FastTransposeSMatrix

main( )

{ TSMatrix a,b;

int k,i,j,v,m,n1,n;

printf( "a数组中行数、列数，非零元的个数");

scanf("%d%d%d",&m,&n1,&n);

for(k=1;k<=n;k++)

{

scanf("%d%d%d",&i,&j,&v);

a.data[k].i=i;

a.data[k].j=j;

a.data[k].v=v;

}

a.mu=m,a.nu=n1,a.tu=n;

FastTransposeSMatrix( [6] ); //函数调用

printf("\n转置后的三元组：\n");

for(k=1;k<=n;k++)

{

printf("%3d%3d%3d\n",b.data[k].i,b.data[k].j,b.data[k].v);

}

}