

noj实验3报告

0.0 题目//需求分析



题目要求求一个三元组表示的矩阵的转置矩阵。

不是直接把第一列和第二列互换就可以的。

应该保证转置后也是按行列有序的。

1.0 实验思路

第一步, 初始化单个三元组和矩阵的三元组表示法

第二步, 在主函数中把非零元素以三元组的形式输进去。

第三步, 调用一次定位快速转置进行操作。

第四步, 一次定位快速转置法, 教材上介绍的清楚, 就是通过一个辅助的位置数组算出每一次转置后的位置, 即可直接变换过去。

2.0 代码

```
//noj1.3 稀疏矩阵的转置
#include<stdio.h>
#include<stdlib.h>
#define MAXSIZE 400
#define OK 1
typedef int Status;
typedef int ElemType;
typedef struct{
```

```

    int row,col;
    ElemType elem;
}Triple;//每个非零元素的三元组
typedef struct{
    Triple data[MAXSIZE+1];//三元组表, data[0]未用
    int m,n,len;//行数, 列数, 非零元素的个数
}TSMatrix;

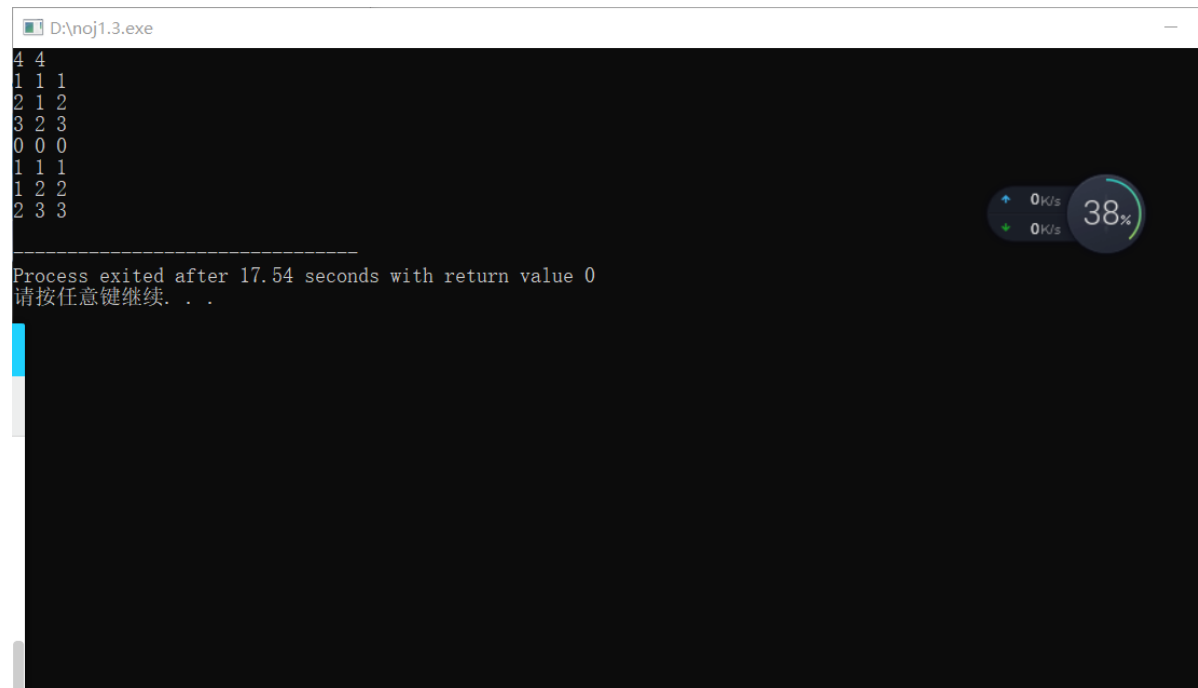
Status FastTransposeTSMatrix(TSMatrix A, TSMatrix *B){
    int col, t, p, q;
    int num[MAXSIZE+1], position[MAXSIZE+1];
    B->len = A.len;
    B->n = A.m;
    B->m = A.n;
    if(B->len){
        //填写num表
        for(col=0; col<A.n; col++){
            num[col] = 0;
        }
        for(t=0; t<A.len; t++){ //一次遍历, 求出num[col]
            num[A.data[t].col]++;
        }
        //填写position表
        position[0] = 0;
        for(col=1; col<A.n; col++){
            position[col] = position[col-1] + num[col-1];
        }
        //将被转置矩阵的三元组表A从头至尾扫描一次, 实现矩阵转置
        for(p=0; p<A.len; p++){
            col = A.data[p].col;
            q = position[col];
            B->data[q].row = A.data[p].col;
            B->data[q].col = A.data[p].row;
            B->data[q].elem = A.data[p].elem;
            position[col]++; //向后移一位
        }
    }
    int i;
    for(i=0; i<B->len; i++){
        printf("%d %d %d", B->data[i].row, B->data[i].col, B->data[i].elem);
        printf("\n");
        //if(i==0)printf("\n");
    }
    return OK;
}

int main()
{
    //把非零元素输进来
    TSMatrix A; //三元组定义
    scanf("%d%d", &A.m, &A.n);
    int k=0;
    int a,b,c;
    while(scanf("%d%d%d", &a, &b, &c)){
        if(a==0 && b==0 && c==0){
            break;//矩阵输入的结束条件
        }
        else{

```

```
        A.data[k].row = a;  
        A.data[k].col = b;  
        A.data[k].elem= c;  
        k++;  
    }  
}  
A.len = k;  
//执行转置操作  
    //首先分配内存  
TSMatrix *B = (TSMatrix*) malloc(sizeof(TSMatrix)); //注意这里要分配内存空间  
FastTransposeTSMatrix(A, B); //一次定位快速转置法  
return 0;  
}
```

3.0 测试结果



```
D:\noj1.3.exe  
4 4  
1 1 1  
2 1 2  
3 2 3  
0 0 0  
1 1 1  
1 2 2  
2 3 3  
-----  
Process exited after 17.54 seconds with return value 0  
请按任意键继续. . .
```

4.0 实验心得

先想思路，再码代码，就有一种按照蓝图做事的感觉，就不会太难。

这也是老师一直教我们的。