

# 数据结构作业4.1/4.2/4.3/4.4/4.12

---

- 姓名：牟鑫一
  - 班级：191174
- 

## 4.1

## 4.2 对称矩阵A的压缩存储方式

(1)

(2)

(3)

## 4.3 合并n阶三角矩阵A、B存到 $n \times n+1$ 阶三交矩阵C

```
1 void Transfer(int *A, int *B) {  
2     int n = 100;          //此处设n=100  
3     int C[100][100+1];  
4     for (int i = 0; i < n; i++) {    //从第一行到第n行赋值  
5         for (int j = 0; j <= i; j++) { //将A数组的第i行元素赋到C数组的第i行  
6             C[i][j] = A[i][j];  
7         }  
8         for (int k = i + 1; k < n; k++) { //将B数组的第i行元素赋到C数组的第i行
```

```

9         C[i][k] = B[k - 1][i];
10     }
11 }
12 }

```

## 4.4

(1)

(2)

### 4.12 查找二元数组A[m][n]的鞍点

```

1  int SaddlePoint(int A[][], int m, int n) {    //传入数组A[m][n]
2      for (int i = 0; i < m; i++) {    //按行循环一次，逐行查找鞍点
3          int temp = A[i][0];    //保存临时“鞍点”值
4          int min = 0;    //记录当前行最小元素列号
5          for (int j = 1; j < n; j++) {    //查找i行的最小元素并记录该元素列号
6              if (A[i][j] < temp) {
7                  temp = A[i][j];
8                  min = j;
9              }
10         }
11         for (int k = 0; k < m; k++) {    //判断i行最小元素是否min列最大元素
12             if (k == i) continue;    //不与自身作比较
13             if (A[k][min] > temp) break;    //只要min列有一个元素比A[i][min]
14             大，就跳出循环继续查找第i+1行
15             if (k == m - 1)    //若程序能执行到这一步说明min列没有比A[i][min]大
16             的元素，A[i][min]即为鞍点
17                 cout << "Find:A[" << i << "][" << min << "]= " << A[i][min]
18                 << endl;    //输出该鞍点
19         }
20     }
21 }

```

函数时间复杂度为：