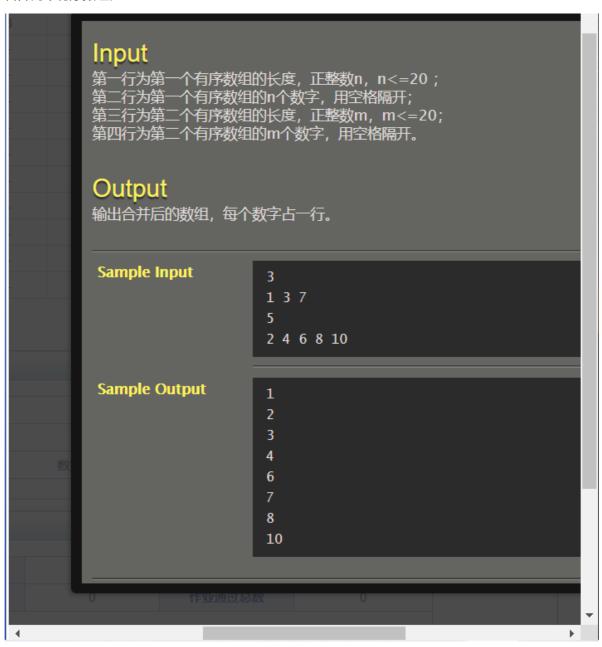
noj实验1报告

0.0 题目

合并两个有序数组。



1.0 代码

```
#include<stdio.h>
int a[20],b[20],c[40];//全局数组
void Merge(int n,int m)
{
   int i=0,j=0,k=0;

   while(i<n && j< m)
   {
      if(a[i] <= b[ j ])
      {
```

```
c[k] = a[i];
        i++;k++;
        else{
        c[k] = b[j]; k++; j++;
    while(i<n)
    c[k] = a[i]; i++;k++;
    while(j<m)</pre>
    c[k] = b[j]; j++;k++;
}
int main()
{
    int i,n,m;
    scanf("%d",&n);
    for(i = 0; i < n ; i++){
    scanf("%d",&a[i]);
    scanf("%d",&m);
    for(i = 0; i < m ; i++){
    scanf("%d",&b[i]);
    Merge( n, m);
    for(i = 0; i < m+n; i++){}
    printf("%d\n",c[i]);
    }
    return 0;
    }
```

2.0 思路

2.1 不一样的地方

我在复习C语言的时候看到了全局变量,这次使用了一下,算是避过了函数间指针的传递。

2.2 简述

和合并链表的思路相似。

创建数组输入输出等省略。在具体比较两个数组时,是三个下标移动之间的关系。

首先设置控制条件使这几个下标要满足他们的调用长度关系,即不能溢出。

当A数组大于B数组,存B当前值入C,BC下标后移一位,A不动。

相同的, B大于A, 存A当前值入C, AC下标后移一位, B不动。

最后的部分,在AB较长者的方面,分A长和B长两种,但操作一致:

把剩下的数组元素都存入C,并且已经能保证整体有序。